KONSTRUKČNÍ ELEKTRONIKA A Radio

ROČNÍK VI/2001. ČÍSLO 3

Amatérské RADIO
ŘADA B - PRO KONSTRUKTÉRY

ROČNÍK L/2001. ČÍSLO 3

V TOMTO SEŠITĚ

KONSTRUKČNÍ ELEKTRONIKA A RADIO

Vydavatel: AMARO spol. s r. o.
Redakce: Radlická 2, 150 00 Praha 5, tel.: (02) 57 31 73 11, tel./fax: (02) 57 31 73 10.
Šéfredaktor ing. Josef Kellner, sekretářka redakce Eva Kelárková, tel. 57 31 73 14.
Ročně vychází 6 čísel. Cena výtisku 36 Kč.
Rozšiřuje PNS a. s., Transpress spol. s r. o., Mediaprint & Kapa a soukromí distributoři.
Předplatné v ČR zajišťuje **Amaro** spol. s r. o. - Michaela Jiráčková, Hana Merglová (Radlická 2, 150 00 Praha 5, tel./fax: (02) 57 31 73 13, 57 31 73 12). Distribuci pro předplatitele také provádí v zastoupení vydavatele společnost Předplatné tisku s. r. o., Abocentrum, Moravské náměstí 12D, P. O. BOX 351, 659 51 Brno; tel: (05) 4123 3232; fax: (05) 4161 6160; abocentrum@pns.cz; reklamace - tel.: 0800-171 181.
Objednávky a predplatné v Slovenskej republike vybavuje MAGNET-PRESS Slovakia s. r. o., Teslova 12, P. O. BOX 169, 830 00 Bratislava 3, tel./fax (07) 444 545 59 - predplatné, (07) 444 546 28 - administratíva; email: magnet@press.sk.
Podávání novinových zásilek povoleno Českou poštou - ředitelstvím OZ Praha (č.j. nov 6005/96 ze dne 9. 1. 1996).
Inzerci v ČR přijímá redakce, Radlická 2, 150 00 Praha 5, tel.: (02) 57 31 73 11, tel./fax: (02) 57 31 73 10.
Inzerci v SR vyřizuje MAGNET-PRESS Slovakia s. r. o., Teslova 12, 821 02 Bratislava, tel./ /fax (07) 44 45 06 93.
Za původnost a správnost příspěvků odpovídá autor (platí i pro inzerci). Nevyžádané rukopisy nevracíme.
http://www.aradio.cz; *E-mail*: pe@aradio.cz
ISSN 1211-3557, MKČR 7443

Dějiny přenosu zpráv na dálku

Historie elektřiny a magnetizmu

Nikola Tesla

Již v roce 1900 chorvatský fyzik a elektrotechnik Nikola Tesla předpověděl v časopise „Century Magazine“, že elektromagnetické vlny se budou využívat k hledání lodí na širém moři. Nikdo tehdy nepředpokládal, že se tato předpověď vyplní již po dvou letech, když Christian Hülsmeyer (1881 až 1957) svými pokusy ukázal, že je to možné, a v principu objevil radarovou techniku.

Nikola se narodil 10. 7. 1856 blízko Gospiče v tehdejší Dalmácii, jeho otec byl knězem ortodoxní církve a určitě nebyl nadšen tím, že jeho syn zanechal teologických studií a začal se zajímat o techniku, zvláště pak o vlny vysokých kmitočtů. Nikola studoval přírovní vědy (dokonce v Praze) a svou praktickou životní dráhu začal u budapešťské firmy Deckert a Homolka, která vyráběla telefonní přístroje. Brzy tam prosadil různá zlepšení, a tak si jej všimli v Edisonově koncernu, který měl pobočku v Paříži. Dostal nabídku, aby vstoupil do jejich společnosti, tu přijal, a v roce 1882 odchází do Paříže.

O dva roky později přesídlil tento erudovaný technik do USA, kde byly laboratoře s dokonalým vybavením, aby tam pokračoval ve svých výzkumech. Nezávisle na Galileo Ferrarisovi (1847 až 1897), a prakticky ve stejné době jako Friedrich August Haselwander (1859 až 1932), objevil v roce 1887 princip třífázového motoru a vícefázové systémy všeobecně. V tomtéž roce obdržel patent na motor pracující bez kolektoru a poté dalších 41 patentů, které využívaly princip točivého elektromagnetického pole. Umožnilo mu to mj. založit vlastní společnost Tesla Electric Company.

Od roku 1890 se však již věnoval vysokofrekvenční technice a o rok později objevil princip Teslova transformátoru, který byl po něm nazván. Jeho úspěchy mu umožnily otevřít si vlastní laboratoř a tak se plně věnovat pokusům v oblastech, které ho zajímaly především. Byl to hlavně bezdrátový přenos energie, do kterého vložil mnoho prostředků i energie.

Postavil vlastní vysílač v Colorado Springs a s jeho pomocí se mu dařilo např. rozsvítit doutnavku na vzdálenost 25 km. V roce 1900 začal se stavbou asi 7x silnějšího vysílače ve Wardenclyfu, ovšem tento projekt se mu nepodařilo dokončit. V následujících letech se věnoval konstrukcím vf přístrojů, a s touto prací vydržel až do své smrti 7. 1. 1943 v New Yorku.

K uctění jeho zásluh byla pojmenována jednotka magnetické indukce 1 T (tesla), 1 T = 1 Vs/m^2. Jeho mozek však zřejmě nepracoval zcela normálně. Ve vzácných chvílích sdílnosti vypravoval o vidinách, které ho pronásledovaly, o světélkujících obludách v temné podmořské říši, které k němu přicházely nejen ve spánku, ale i za bílého dne. Vymýšlel proti nim s vypětím vůle stroje a zařízení, jimiž je od sebe odháněl.

Podle vlastních tvrzení měl jasnovidecké schopnosti, dokázal vidět ve tmě a snížit svůj puls na několik tepů za minutu. Stroj si dokázal v duchu nejen navrhnout, ale i poznat, jak bude fungovat - v duchu prý zcela jasně viděl, zda běží pravidelně, či zda je ještě něco třeba vylepšit. I slabé zvuky slyšel přes několik zdí. Prohlašoval, že zachytil šifrované zprávy pocházející z kosmu, domníval se, že z Marsu.

Pracoval i na vojensky využitelných objevech, o kterých se ví velmi málo, a armáda ještě po jeho smrti během druhé světové války zabavila důležité části jeho poznámek.

Existují dokonce teorie, že se jednalo o mimozemšťana, který se dostal na Zem, a zde, s našimi omezenými technickými prostředky, se snaží zrealizovat jinde užívané technologie. Např. jeho „Světová stanice“, jejíž projekt začal realizovat na Long Islandu, už sledovala i oficiálně zcela utopické cíle. Tesla prohlašoval - a bylo to prosím již v roce 1900!, že vytvoří vysílací středisko, které umožní spojení s celým světem nejen pro vysílání politických, obchodních a soukromých zpráv, hudby a zábavných pořadů, ale že jeho stanice umožní i přenosy utajených rozhovorů, obrazů, fotografií a dokumentů. Chtěl odtamtud určovat polohu všech lodí na mořích na celém světě a navazovat s nimi spojení. Kromě toho odtud chtěl dodávat světu i samotnou energii. Výstavbu nedokončil proto, že jeho mecenáši zastavili finanční podporu.

Literatura

[1] Wireless Engineer 1/1923, London.
[2] *Tesla, N.*: Moji pronalasci. Zagreb 1977.
[3] *Cheney, M.*: Tesla, Man Out of Time. New Jersey 1981.
[4] Nikola Tesla. Sborník k 80. výročí narození. Beograd 1936.

Ing. Jiří Peček, OK2QX

STAVEBNICE S MIKROPOČÍTAČI ŘADY „51“

Jiří Kadlec

Ačkoli se v dnešní době uvedly na trh mikropočítače PIC, řada mikropočítačů „51” stále zaujímá významné místo na trhu. Mikropočítače řady „51” nahradily zastaralou řadu „48” a v poslední době prošly značným vývojem až po řadu 89Cxx s integrovanou pamětí dat typu FLASH na čipu. Významným krokem k tomu přispěla firma ATMEL. Nejrozšířenější typy pro standardní aplikace v řadě 80C51 jsou typy 80C31, 87C51 a 89C51 a typy s rozšířenou pamětí programu 80C32, 87C52 a 89C52. Pro velmi náročné aplikace to jsou 80C552 a 87C552 a pro nenáročné aplikace mikropočítače 89C2051 s programovou pamětí typu FLASH na čipu a v pouzdru DIL20.

V tomto čísle KE je popsána stavebnice řídicího systému s mikropočítačem 80C552, která obsahuje základní desku BASIC 552, desku klávesnice a displeje KEYDSP1 a desku PORT 64 pro rozšíření počtu portů. Desky s plošnými spoji i celé sady součástek je možné objednat u autora na telefonním čísle, uvedeném v závěru za konstrukční částí. Za konstrukční částí následují kapitoly věnované podrobnému popisu funkce procesorů 80C51 a 80C552 a souboru instrukcí mikropočítačů řady „51”.

Při popisu vývodů IO jsou v článku uváděny jak anglické, tak evropské indexy. To znamená, že např. napájecí vstupy mohou být označovány jako V (voltage) nebo jako U. Taktéž popis některých vývodů nemusí odpovídat katalogovým údajům, protože různí výrobci označují shodné vývody obvodů odlišně (např. OC nebo OE).

V následujícím přehledu jsou uvedeny standardní typy mikropočítačů řady „51” se základními charakteristikami:

TYP	ROM/EPROM	RAM	RYCHL.	POUZDRO	FUNKCE	POZN.
80C31	0	128	33 MHz	DIL40, LCC44, QFP44	UART, 2 časovače	
80C51	4 k ROM	128	33 MHz	DIL40, LCC44, QFP44	UART, 2 časovače	
87C51	4 k EPROM	128	33 MHz	DIL40, LCC44, QFP44	UART, 2 časovače	
89C51	4 k FLASH	128	33 MHz	DIL40, LCC44, QFP44	UART, 2 časovače	ATMEL
83C51FA	8 k ROM	256	24 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	
87C51FA	8 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	
83L51FA	8 k ROM	256	20 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	3 až 4,5 V Ucc
87L51FA	8 k EPROM	256	20 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	3 až 4,5 V Ucc
87C51FB	16 k ROM	256	24 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	
83C51FB	16 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	
87L51FB	16 k ROM	256	20 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	3 až 4,5 V Ucc
83L51FB	16 k EPROM	256	20 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	3 až 4,5 V Ucc
87C51FC	32 k ROM	256	24 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	
83C51FC	32 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA	
80C32	0	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
80C52	8 k ROM	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
87C52	8 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
89C52	8 k FLASH	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	ATMEL
80C54	16 k ROM	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
87C54	16 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
80C58	32 k ROM	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
87C58	32 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	UART, 3 časovače	
80C451	0	128	16 MHz	DIP64, LCC68	UART, 2 čas., ext. I/O	
83C451	4 k ROM	128	16 MHz	DIP64, LCC68	UART, 2 čas., ext. I/O	
87C451	4 k EPROM	128	16 MHz	DIP64, LCC68	UART, 2 čas., ext. I/O	
83C504	16 k ROM	256	24 MHz	DIL40, LCC44, QFP44	24/8 dělič., 2 čas.	
87C504	16 k EPROM	256	24 MHz	DIL40, LCC44, QFP44	24/8 dělič., 2 čas.	
87C524	16 k EPROM	512	20 MHz	DIL40, LCC44, QFP44	UART, 3 čas., WD, I2C	
83C524	16 k ROM	512	12 MHz	DIL40, LCC44, QFP44	UART, 3 čas., WD, I2C	
83C528	32 k ROM	512	16 MHz	DIL40, LCC44, QFP44	UART, 3 čas., WD, I2C	
87C528	32 k EPROM	512	16, 20 MHz	DIL40, LCC44, QFP44	UART, 3 čas., WD, I2C	
83CE528	32 k ROM	512	16 MHz	QFP	UART, 3 čas., WD, I2C	

80C550	0	128	16 MHz	LCC44, DIL40	UART, 2 čas., WD, ADC8	
83C550	4 k ROM	128	16 MHz	LCC44, DIL40	UART, 2 čas., WD, ADC8	
87C550	4 k EPROM	128	16 MHz	LCC44, DIL40	UART, 2 čas., WD, ADC8	
80C552	0	256	16, 24 MHz	LCC68, QFP80	UART, 2 čas., CPT, I2C, 2x PWM, ADC10 in	
83C552	8 k ROM	256	16, 24 MHz	LCC68, QFP80	UART, 2 čas., CPT, I2C, 2x PWM, ADC10 in	
87C552	8 k EPROM	256	16 MHz	LCC68, QFP80	UART, 2 čas., CPT, I2C, 2x PWM, ADC10 in	
83CE558	32 k ROM	1k	16 MHz	QFP80	jako 80C552 s PLL, auto scan ADC	
89CE558	32 k FLASH	1k	16 MHz	QFP80	jako 80C552 s PLL, auto scan ADC	
80CE558	0			QFP80	jako 80C552 s PLL, auto scan ADC	
80C562	0	256	16 MHz	LCC68, QFP80	UART, 2 čas., CPT, 2x PWM, ADC8 in.	
83C562	8 k ROM	256	16 MHz	LCC68, QFP80	UART, 2 čas., CPT, 2x PWM, ADC8 in.	
80C575	0	256	16 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA, 4x analog. kompar.	
83C575	8 k ROM	256	16 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA, 4x analog. kompar.	
87C575	8 k EPROM	256	16 MHz	DIL40, LCC44, QFP44	en. UART, 3 čas., PCA, 4x analog. kompar.	
83C576	8 k ROM	256	16 MHz	DIL40, LCC44, SDIL42	10 bit AD, 3 čas., PCA, WD	
87C576	8 k EPROM	256	16 MHz	DIL40, LCC44, SDIL42	10 bit AD, 3 čas., PCA, WD	
80C592	0	512	16 MHz	LCC68, QFP80	8xC552 + CAN interf.	
83C592	16 k ROM	512	16 MHz	LCC68, QFP80	8xC552 + CAN interf.	
87C592	16 k EPROM	512	16 MHz	LCC68, QFP80	8xC552 + CAN interf.	
83CE598	32 k ROM	512	16 MHz	QFP80	8xC552 + CAN, bez I2C	
87CE598	32 k EPROM	512	16 MHz	QFP80	8xC552 + CAN, bez I2C	
80CE598	0	512	16 MHz	QFP80	8xC552 + CAN, bez I2C	
80C652	0	256	16, 24 MHz	DIL40, LCC44, QFP44	UART, 2 čas., I2C	
83C652	8 k ROM	256	16, 24 MHz	DIL40, LCC44, QFP44	UART, 2 čas., I2C	
87C652	8 k EPROM	256	16, 24 MHz	DIL40, LCC44, QFP44	UART, 2 čas., I2C	
83C654	16 k ROM	256	16, 24 MHz	DIL40, LCC44, QFP44	UART, 2 čas., I2C	
80C654	0	256	16, 24 MHz	DIL40, LCC44, QFP44	UART, 2 čas., I2C	
83CE654	16 k ROM	256	16 MHz	QFP44	UART, 2 čas., I2C	
80CE654	0	256	16 MHz	QFP44	UART, 2 čas., I2C	
83C750	1 k ROM	64	40 MHz	SDIP24	1 časovač	
87C750	1 k EPROM	64	40 MHz	SDIP24	1 časovač	
83C751	2 k ROM	64	16 MHz	DIP24, LCC28	1 čas., I2C	
87C751	2 k EPROM	64	16 MHz	DIP24, LCC28	1 čas., I2C	
83C752	2 k ROM	64	16 MHz	DIP24, LCC28	1 čas., I2C, PWM, ADC	
87C752	2 k EPROM	64	16 MHz	DIP24, LCC28	1. čas., I2C, PWM, ADC	
80C851	0	128	16 MHz	DIL40, LCC44, QFP44	UART, 2 čas.	
83C851	4 k ROM	128	16 MHz	DIL40, LCC44, QFP44	UART, 2 čas.	
83C852	6 k ROM	256	6 MHz			
83C055	16 k ROM	256	12 MHz	DIP42		
87C055	16 k EPROM	256	12 MHz	DIP42		
89C1051	1 k FLASH	128	24 MHz	DIP20, SOIC20	UART, 2 časovače	ATMEL
89C2051	2 k FLASH	128	24 MHz	DIP20, SOIC20	UART, 2 časovače	ATMEL

Základní charakteristiky mikropočítačů řady „51"

1. Dokáží adresovat 64 k vnější paměti programu.
2. Dokáží adresovat 64 k vnější paměti dat.
3. Osmibitová ALU.
4. Oscilátor a generátor hodinového kmitočtu integrovaný na čipu.
5. Dva šestnáctibitové čítače/časovače (mikropočítače typů 8x52 mají tři časovače).
6. Dvě prioritní úrovně pro pět zdrojů žádosti o přerušení (pro šest zdrojů žádosti o přerušení u typů 8x52).
7. Plně duplexní sériový kanál.
8. Režim sníženého příkonu.
9. Speciální funkční registry SFR.

Mikropočítače 80C552 obsahují navíc:

1. Desetibitový AD převodník s osmi multiplexovanými vstupy.
2. Dva osmibitové výstupy s pulsně šířkovou modulací.
3. Pět osmibitových IO portů.
4. Sběrnici I2C.
5. Hlídací obvod WATCHDOG.

BASIC-552 - základní deska s mikropočítačem 80C552

Jedním z nejmocnějších mikropočítačů řady „51" je typ 80C552, který je použit v popisované stavebnici řídicího systému.

Stavebnice je tvořena základní deskou BASIC-552, kartou klávesnice a displeje KEYDSP1 a kartou PORT 64 pro rozšíření počtu portů.

Řídicí systém se uplatní při řízení různých technologických procesů, jako je ovládání strojů, řízení diskotékových světel, řízení křižovatek, sběr a vyhodnocování jakýchkoliv dat apod.

Přes rozhraní RS232 může systém komunikovat s počítačem PC a přes rozhraní I2C s kterýmikoliv zařízeními, obsahujícími taktéž sběrnici I2C.

Popis zapojení

Schéma základní desky BASIC-552 je na obr. 1. Zařízení je sestaveno tak, aby uspokojilo kromě amatérských konstruktérů i náročné uživatele.

Základní deska obsahuje mikropočítač, adresový záchytný registr, paměť programu 64 kbyte, paměť dat se dvěma paměťovými obvody (lze zvolit paměti SRAM, EEPROM 2x 32 kbyte nebo jejich kombinaci), sériovou paměť EEPROM, adresový dekodér, hlídací obvod napájení se zálohovaným napětím z baterie, obvod reálného času RTC, obvod sériového styku RS232, zdroj a konektory pro připojení externích desek (klávesnice, displeje, stykových obvodů interface atd.).

Mikropočítač 80C552 obsahuje osm multiplexovaných vstupů ADC0 až ADC7 desetibitového A/D převodníku, sériový styk I2CBUS, sériový styk UART, výstupy PWM pro impulsní řízení, vstupy pro externí přerušení, hardwarový hlídací obvod WATCHDOG a časovač T2, který je možné ovládat vnějšími signály CT, T2 a RT2.

Hlavními součástmi základní desky jsou zdrojová část, mikropočítač 80C552, adresový záchytný registr IO2 a paměť programu IO3. Tyto čtyři základní součásti tvoří jádro systému, které je schopné spolu s okolními pasivními součástkami pracovat samostatně.

Jako mikropočítač lze použít typ 80C552 bez interní paměti dat nebo typ 87C552 s pamětí dat EPROM. Způsob čtení instrukcí z datové paměti jednotlivých typů je popsán v odstavci Paměť programu. Mikropočítač vykonává instrukce, které jsou uložené v programové paměti na jednotlivých adresách. U mikropočítače 80C552 je způsob adresování takový, že nejprve je na datovou sběrnici P0 vystaven nižší byte adres A0 až A7, který je signálem ALE zapsán do adresového záchytného registru IO2. Pak je na port P2 vystaven vyšší byte adresy A8 až A15. Tímto způsobem je potom přítomna celá adresa A0 až A15 na adresové sběrnici.

Obr. 1. BASIC-552 - základní deska s mikroprocesorem 80C552

V součinnosti se signálem PSEN (non) je přečtena instrukce z externí paměti programu IO3. Signály ALE a PSEN (non) jsou prvními signály, jejichž přítomnost ověřujeme při oživování systému.

Kmitočet oscilátoru mikropočítače je řízen krystalem 11,0592 MHz a je určen s ohledem na přenosovou rychlost sériového styku RS232. S uvedeným krystalem lze s časovačem 1 generovat standardní přenosové rychlosti 62,5 kHz, 9,6 kHz, 4,8 kHz, 2,4 kHz nebo 1,2 kHz. Lze však osadit krystal s jakýmkoliv kmitočtem od 1,2 do 16 MHz pro typ 87C552, popř. od 1,2 do 24 MHz (30 MHz) pro typ 80C552. S krystalem 6 MHz lze uskutečnit přenosovou rychlost 0,11 kHz, při použití krystalu 12 MHz pak rychlosti 0,11 kHz, 187,5 kHz, 375 kHz a 1 MHz. Mezní kmitočty krystalu je nutné dodržet podle podmínek v katalogových údajích výrobců mikropočítačů. Kondenzátory C3 a C4 zabezpečují start a stabilitu oscilátoru.

Každý port mikropočítače se může chovat buď jako běžný vstupně/výstupní port nebo může vykonávat alternativní funkci. Tak např. port P0 může být využit u mikropočítače 87C552 s interní pamětí programu čistě jako vstupně/výstupní port (pokud není zapotřebí adresovat bit A0 vnější adresové sběrnice). U mikropočítače 80C552, který interní paměť programu nemá, musí port P0 vykonávat multiplexovanou funkci adresování a práce s daty (A0, D0).

V zapojení na obr. 1 je testovací port P34/ /T0 využit jako výstupní pro generování adresy A16, která v podstatě rozlišuje mezi spoluprací s pamětmi dat (SRAM nebo EEPROM) a ostatních obvodů (RTC a vně připojených). Podle potřeby je nutno tento port ovládat softwarově. Pokud vyvstane požadavek využívat testovací vstup T, lze použít k této funkci port P35/T1, avšak pouze v případě, že není využit jako druhý výstup pro sériový styk se signálem CTS.

Pro správnou činnost A/D převodníků je nutno zajistit přesné referenční napětí na vstupech AVREF+ a AVREF-. O to se stará obvod IO12 typu LM336, což je referenční dioda 2,5 V. Protože vstup AVREF- je připojen na napájecí zem GND, jsou k tomuto potenciálu vztaženy všechny vstupy desetibitového AD převodníku v mikroprocesoru. Namísto referenční diody IO12 lze použít diodu s jiným napětím až do maximální velikosti 4,5 V. I když je možné na vstup AVREF+ připojit referenční napětí až 5 V, v žádném případě nesmí být tento vstup propojen s napájecím napětím Vcc. Napájecí větev nemusí být stabilní (mohou se na ní vyskytovat rušivé signály) a hodnota po ukončení převodu by byla značně zkreslená. Kondenzátor C8 filtruje napětí za referenční diodou. Vývody AVSS a AVDD mikroprocesoru jsou určeny pro napájení interního AD převodníku a jsou připojeny na napájecí větev GND a Vcc.

Napájecí napětí z externího zdroje se přivádí na konektor X1, za kterým následuje diodový můstek D2 až D5, filtrační kondenzátory C5, C12 a C13 a stabilizátor IO9. Za stabilizátorem je napětí filtrováno kondenzátory C6 a C14. V napájecí větvi na desce jsou u jednotlivých obvodů rozmístěny filtrační keramické kondenzátory hodnoty 100 nF.

Paměť programu

Obvod IO3 slouží jako paměť programu. Typ 27C512 (EPROM) zajišťuje plnou kapacitu 64 kbyte, kterou lze adresovat mikropočítačem 80C552. V součinnosti se signálem PSEN\ jsou aktivovány její vstupy CE\ a OE\. V případě nutnosti simulovat program lze do patice IO3 zasunout simulátor EPROM. Signál PSEN\ je generován automaticky mikropočítačem.

Pokud je použit mikropočítač 80C552, který neobsahuje interní paměť programu, musí být signál EA\ spojen s napájecí zemí GND. Pokud je použit typ 87C552 s interní EPROM 8 kbyte, spojením EA\ s Vcc čte mikropočítač instrukce z interní EPROM od adresy 0 do adresy 1FFFH a od adresy 2000H do FFFFH čte instrukce z externí EPROM. Adresy od 0003H do 0073H jsou využity pro obsluhu přerušení.

Paměť dat

Paměť dat tvoří obvody IO4 a IO5. Do objímek lze vložit statické SRAM typu 62256, EEPROM typu 28C256 nebo jejich kombinace. Propojky J1 a J2 určují, jak bude příslušná paměť napájena. V případě použití statické SRAM je nutné propojit napájení se zálohovaným napětím z baterie VCUB, v případě použití EEPROM je paměť nutno napájet přímo napájecím napětím 5 V.

Obě paměti IO4 a IO5 jsou adresovány bity A0 až A14. Pouze bit A15, který je připojen k dekodéru adres IO6 rozhoduje, zda v daném okamžiku bude pracováno s pamětí IO4 nebo IO5. Ještě záleží na stavu adresového bitu A16, který je generován portem P34/T0 mikropočítače a musí být v úrovni log. 1 (tento stav je nastaven automaticky po vynulování mikroprocesoru).

Funkci si popíšeme přímo na vysílaných adresách. Od adresy 10000H až do 13FFFH je na výstupu 1Y0 dekodéru IO6 úroveň log. 0, čímž je uvolněn CE\ paměti IO4. Od adresy 14000H do adresy 17FFFH je na výstupu 1Y1 dekodéru IO6 úroveň log. 0. Výstup 1Y1 je však spojen s výstupem 1Y0 (na výstupech adresového dekodéru IO6 jsou otevřené kolektory tranzistorů), takže stále je adresována paměť IO4. Adresování paměti IO4 probíhá v rozsahu adres 10000H až 17FFFH. Od adresy 18000H je aktivován bit A15 a na dekodéru IO6 je výstup 1Y2 v úrovni log. 0. V tomto stavu liž není aktivován CE\ paměti IO4, ale je aktivován CE\ paměti IO5. Protože má dekodér IO6 spojené výstupy 1Y2 a 1Y3, probíhá adresace paměti IO5 v rozsahu adres 18000H až 1FFFFH.

Zápis a čtení pamětí dat je řízeno mikropočítačem, lze využít instrukce assembleru MOVX. Protože mikropočítač je osmibitový, může být pro adresování využit šestnáctibitový registr DPTR. Po nastavení P34 do úrovně log. 1 se vykonáním instrukce MOVX @DPTR,A přesune byte dat z akumulátoru do externí paměti dat, nepřímo adresovatelné registrovým párem DPTR. Při tomto způsobu adresování je k dispozici celý prostor 64 kbyte. Před tím je nutné naplnit registr DPTR šestnáctibitovou adresou tak, že nižší byte adresy (A0 až A7) se uloží do akumulátoru a následně přesune do DPL instrukcí MOV DPL,A a vyšší byte adresy (A8 až A15) se stejným způsobem zapíše instrukcí MOV DPH,A. Alternativně instrukce MOVX A, @DPTR přesune byte z externí paměti dat do akumulátoru A. Jiným způsobem adresování je využít registry R0 a R1, kterými však lze adresovat pouze prostor do 256 byte.

Po instrukci zápisu může u pamětí EEPROM trvat (díky pomalému hardware) vlastní zápis až 10 ms, s čímž musí programátor počítat, a buď musí vyčkat před dalším zápisem, nebo může využít čtení příznaku BUSY z paměti.

Protože při výpadku napájení (kromě zálohovaného napětí) musí být na vstupech CE\ úroveň log. 1, jsou rezistory R4 a R5 připojeny na zálohované napětí VCUB.

Sériová paměť EEPROM

Na pozici IO11 lze do objímky vložit sériovou paměť EEPROM typu 93C46 o velikosti 128x 8 bitů nebo 93C56 o velikosti 256x 8 bitů. Tyto paměti mohou být organizovány pro práci s osmi nebo šestnáctibitovými daty. Tím, že je vstup ORG paměti spojen se zemí, je zvolena organizace pro práci s osmibitovými daty.

Sériové EEPROM lze využít např. v případě, když jsou na pozicích IO4 a IO5 osazeny SRAM a je požadováno pracovat s daty, která musí být dostupná po výpadku napájení i záložní baterie. Do sériové EEPROM lze zapisovat až 100000x a data mohou být uchována po dobu 100 let (typ CAT93C46 nebo CAT93C56).

Nyní si popíšeme práci s pamětí 93C56.

S pamětí se pracuje tak, že nejprve signál CS přejde do úrovně log. 1 a v součinnosti signálů CLK a DI je do paměti zapsána instrukce, která udává, co se následně bude dít. Následuje práce s daty (pokud je k tomu instrukce určena). Na závěr je nutno uvést signál CS do úrovně log. 0.

Vlastní instrukci tvoří start bit, operační kód a adresa. Instrukcí, se kterými paměť může pracovat, je celkem sedm. Každá instrukce se skládá z dvanácti bitů (z jedenácti bitů v případě organizace 128x 16). Instrukce, adresy a zapisovaná data jsou zpracovávána náběžnou hranou hodinového signálu CLK. Výstup dat DO má v klidovém stavu vysokou impedanci, a proto je možné jej spojit se vstupem dat DI.

Pokud poklesne napájecí napětí pod 3,5 V, je vnitřně v paměti zakázáno mazat a zapisovat data do paměti a instrukce EWDS (zákaz zápisu) je provedena automaticky interně v obvodu.

V tab. 1 je uveden instrukční soubor použité sériové EEPROM.

Čtení dat READ. Procesor umožňuje číst data z paměti na zvolené adrese. Mezi čtením, mazáním a zápisem dat se rozhoduje pouze operačním kódem, takže je nutné dát si pozor na zápis správného operačního kódu.

Nejprve je nutné uvést CS do úrovně log. 1. Potom na DI (P42 u mikropočítače) se vystaví úroveň log. 1 a náběžnou hranou na CLK (P41) se zapíše start bit. Stejným způsobem se zapíší postupně bity operačního kódu 10. Následuje zápis jednotlivých bitů adresy od A8 do A0. Po celou dobu je výstup DO z paměti ve stavu vysoké impedance a nyní přechází do úrovně log. 0. Následuje čtení dat D7 až D0 z určené adresy na výstup DO (P42) v součinnosti náběžných hran impulsů na CLK (P41). Na závěr musí CS přejít do úrovně log. 0.

Uvolnění zápisu EWEN (Erase Write Enable). Před každým zápisem dat do paměti je nutné provést tuto operaci. Pokud uvolnění zápisu nebude povoleno, celý zápis do paměti je ignorován. Funkce EWEN je velice vhodná k tomu, aby zabránila nechtěnému zápisu dat do paměti v době, kdy je na mikroprocesoru sníženo napájecí napětí a mikroprocesor může vykonávat náhodné operace. Při poklesu napájecího napětí pod 3,5 V je operace EWEN automaticky vykonána. Po provedení EWEN musí být CS v úrovni log. 0 minimálně 250 ns.

Nejprve je nutné uvést CS do úrovně log. 1. Potom se na DI vystaví úroveň log. 1 a náběžnou hranou na CLK se zapíše start bit. Stejným způsobem zapisujeme postupně bity operačního kódu 00. Nyní je nutno zapsat adresu 11xxxxxxx. Na pozice x lze zapsat libovolná data. Na závěr musí CS přejít do úrovně log. 0.

Zákaz zápisu EWDS (Erase Write Disable). Tato operace by měla být vykonána po každém zápisu. Pokud bude vykonána, paměť zápis neuvolní a proces následného zápisu bude ignorován.

Nejprve je nutné uvést CS do úrovně log. 1. Potom se na DI vystaví úroveň log. 1 a náběžnou hranou na CLK se zapíše start bit. Stejným způsobem zapisujeme postupně bity operačního kódu 00. Nyní je nutno zapsat adresu 00xxxxxxx. Na pozice x lze zapsat libovolná data. Na závěr musí CS přejít do úrovně log. 0.

Mazání dat na zvolené adrese ERASE. Operace vymaže data na zvolené adrese. Po vykonání ERASE musí být CS v úrovni log. 0 minimálně 250 ns.

Nejprve je nutné provést celý proces uvolnění zápisu, na jehož konci se uvede CS do úrovně log. 0. Potom CS se znovu uvede do

úrovně log. 1. Na DI se vystaví úroveň log. 1 a náběžnou hranou na CLK se zapíše start bit. Stejným způsobem se zapíší postupně bity operačního kódu 11. Následuje zápis jednotlivých bitů adresy od A8 do A0. Na závěr této procedury musí CS přejít do úrovně log. 0. Pokud bezprostředně po této akci uvedeme CS do úrovně log. 1 a čteme DO (port P42), přechodem DO do úrovně log.1 oznamuje paměť, že skončila proces zápisu a je připravena na další operaci. CS se nakonec uvede do úrovně log. 0. Nyní by měl následovat celý proces zákazu zápisu EWDS.

Zápis dat na zvolenou adresu WRITE. Nejprve je nutné provést celý proces uvolnění zápisu, který je ukončen uvedením CS do úrovně log. 0. Potom CS se znovu uvede do úrovně log. 1. Na DI se vystaví úroveň log. 1 a náběžnou hranou na CLK se zapíše start bit. Stejným způsobem se zapíší postupně bity operačního kódu 01. Následuje zápis adresy od A8 do A0 a dat od D7 do D0. Na závěr této procedury musí CS přejít do úrovně log. 0. Pokud je bezprostředně po této akci uveden CS do úrovně log. 1 a čten DO (port P4.2), přechodem DO do úrovně log. 1 oznamuje paměť, že skončila proces zápisu a je připravena na další operaci. Na závěr se CS uvede do úrovně log. 0. Nyní by měl následovat celý proces zákazu zápisu EWDS.

Mazání celého paměťového prostoru ERAL (Erase All). Nejprve je nutné provést celý proces uvolnění zápisu, na jehož konci se uvede CS do úrovně log. 0. Potom se CS znovu uvede do úrovně log. 1. Na DI se vystaví úroveň log. 1 a náběžnou hranou na CLK se zapíše start bit. Stejným způsobem se zapíší postupně bity operačního kódu 00. Nyní je nutno zapsat adresu 10xxxxxxx. Na pozice x lze zapsat libovolná data. Na závěr musí CS přejít do úrovně log. 0. Pokud bezprostředně po této akci je uveden CS do úrovně log. 1 a čten DO (port P4.2), přechodem DO do úrovně log. 1 oznamuje paměť, že skončila proces zápisu a je připravena na další operaci. Nakonec CS uvedeme do úrovně log. 0. Nyní by měl následovat celý proces zákazu zápisu.

Zápis dat do celého paměťového prostoru WRAL (Write All). Tímto způsobem budou zapsána stejná data do celého paměťového prostoru. Nejprve je nutné provést celý proces uvolnění zápisu, na jehož konci se uvede CS do úrovně log. 0. Potom se uvede CS do úrovně log. 1. Na DI se vystaví úroveň log. 1 a náběžnou hranou na CLK se zapíše start bit. Stejným způsobem se zapíší postupně bity operačního kódu 00. Nyní je nutno zapsat adresu 01xxxxxxx. Na pozice x lze zapsat libovolná data. Následuje zápis dat D7 až D0. Pokud bezprostředně po této akci uvedeme CS do úrovně log. 1 a čteme DO (port P4.2), přechodem DO do úrovně log. 1 oznamuje paměť, že skončila proces zápisu a je připravena na další operaci. Na závěr se CS uvede do úrovně log. 0. Nyní by měl následovat celý proces zákazu zápisu.

RTC

IO10 typu RTC72421 je obvod reálného času RTC (real time clock). Obsahuje krystalový oscilátor s děličkou, registry sekund, minut, hodin, dní, měsíců, roků a týdnu v měsíci. Mimo jiné obsahuje registry D, E a F, které slouží k přednastavení obvodu nebo hlásí jeho stav. Do obvodu je možné zapisovat data (nastavovat čas nebo registry) nebo je zpětně číst. Po vhodném nastavení registrů obvod sám čítá čas, který je možné číst, nebo oznamuje výstupem STD (sestupnou hranou nebo úrovní pro přerušení mikroprocesoru) uplynutí přednastavené doby. Výstup STD je možno propojkou na konektoru J7 propojit s portem P3.2 nebo P3.3 mikroprocesoru, čímž je možno vyvolat hardwarový INTERRUPT0 nebo INTERRUPT1. Je však nutné zajistit, aby přerušení nekolidovalo s přerušením od obvodu IO8, který hlídá napájecí napětí.

V zapojení není využito zápisu adresy do adresového záchytného registru, a proto je vývod ALE připojen na výstup Uout obvodu IO8. Toto propojení je nutné pro uchování dat a správnou činnost obvodu při napájení ze záložní baterie a jeho význam bude popsán dále.

V tab. 2 jsou uvedena čísla registrů a data, která se nacházejí na příslušných adresách.

Dále je vysvětlen význam jednotlivých zkratek v tab. 2.

PM/AM - v modu 12h může být bit nastaven. Ve 24h modu je ignorován.

30s ADJ - po nastavení tohoto bitu do stavu log. 1 obvod sám vynuluje tento bit po 125 ms. V této době není obvod možné číst, ani do něj zapisovat.

IRQ FLAG - indikuje, zda je generována žádost o přerušení a odpovídá negovanému stavu výstupu STD.

BUSY - stavový bit připraven.

HOLD - po nastavení tohoto bitu může být čten BUSY. Pokud je CS1 ve stavu log. 0, potom i HOLD je ve stavu log. 0.

t1, t0 - nastavení periody pro výstup přerušení STD. Hodnota 0D znamená žádost o přerušení každých 15,6 ms, 1D každou 1s, 2D každou minutu a 3D každou hodinu.

ITRP/STDN - ve stavu log. 0 je povoleno přerušení podle doby, nastavené bity t1, t0.

MASK - maskuje výstup STD.

TEST - nesmí se nastavovat do stavu log. 1, slouží k testování obvodu (přepíná vnitřní děličku na vyšší hodinovou frekvenci).

24/12 - nastavuje mod čítání. Stavem log. 1 se nastaví cyklus 24 h, log. 0 nastaví cyklus 12 h. Nutno nastavovat při REST ve stavu log. 1.

STOP - zastavuje generování impulsů na STD.

REST - zastavuje a nuluje vnitřní děličku.

Tab. 2 je pouze informativní a neudává úplný přehled činnosti obvodu (což není předmětem článku). Pro úplný přehled funkcí je nutné opatřit odpovídající dokumentaci k obvodu.

Zápis dat do obvodu. Data jsou zapisována součinností datových signálů D0 až D3, adresových signálů A0 až A3 a signálů CS2 a WR\.

Mikropočítač vykoná obyčejnou funkci zápisu svým signálem WR\. Je však zřejmé, že data budou pouze čtyřbitová. Není přípustné zapisovat data, která jsou mimo rozsah možných hod-

Tab. 1. Instrukční soubor použité sériové EEPROM typu 93C56.

Instrukce	Bit start	Op. kód	Adresa	Data	Popis
READ	1	10	A8 až A0		čtení z paměti na zvolené adrese
ERASE	1	11	A8 až A0		mazání dat na zvolené adrese
WRITE	1	01	A8 až A0		zápis dat na zvolenou adresu
EWEN	1	00	11xxxxxxx	D7 až D0	uvolnění zápisu
EWDS	1	00	00xxxxxxx		zákaz zápisu
ERAL	1	00	10xxxxxxx		mazání celé paměti
WRAL	1	00	01xxxxxxx	D7 až D0	zápis dat do celého pam. prostoru

Tab. 2. Čísla registrů a data, která se nacházejí na příslušných adresách v RTC typu RTC72421.

Č. reg.	Adresa				Data				Možná hodnota	Popis
	A3	A2	A1	A0	D3	D2	D1	D0		
0	0	0	0	0	s8	s4	s2	s1	0 až 9	jednosekundový reg.
1	0	0	0	1	-	s40	s20	s10	0 až 5	desetisekundový reg.
2	0	0	1	0	mi8	mi4	mi2	mi1	0 až 9	jednominutový reg.
3	0	0	1	1	-	mi40	mi20	mi10	0 až 5	desetiminutový reg.
4	0	1	0	0	h8	h4	h2	h1	0 až 9	jednohodinový reg.
5	0	1	0	1	-	PM/AM	h20	h10	0 až 2 (1)	PM/AM, desetihod. reg.
6	0	1	1	0	d8	d4	d2	d1	0 až 9	jednodenní reg.
7	0	1	1	1	-	-	d20	d10	0 až 3	desetidenní reg.
8	1	0	0	0	m8	m4	m2	m1	0 až 9	jednoměsíční reg.
9	1	0	0	1	-	-	-	m10	0 až 1	desetiměsíční reg.
A	1	0	1	0	r8	r4	r2	r1	0 až 9	jednoroční reg.
B	1	0	1	1	r80	r40	r20	r10	0 až 9	desetiroční reg.
C	1	1	0	0	-	w4	w2	w1	0 až 6	týdenní registr
D	1	1	0	1	30s ADJ	IRQ FLAG	BUSY	HOLD	-	řídicí registr D
E	1	1	1	0	t1	t0	ITRP/STDN	MASK	-	řídicí registr E
F	1	1	1	1	TEST	24/12	STOP	REST	-	řídicí registr F

Vysvětlivky k tabulce:
0 = log. 0; 1 = log. 1; REST = RESET; ITRP/STDN = INTERRUPT/STANDARD; s = sekunda; mi = minuta; h = hodina; d = den; m = měsíc; r = rok; w = týden

not (možné hodnoty jsou v tab. 2). Pokud budou zapsána špatná data, obvod nemusí správně pracovat.

Data zapisujeme instrukcí MOVX @DPTR,A, přičemž čtyřbitová data musí být nejprve zapsána do akumulátoru A v pořadí od LSB (nejméně významový bit D0 až D3), horní čtyři bity D4 až D7 mohou nabývat jakékoliv hodnoty a nemají na vlastní zápis vliv. Adresa se skládá z přednastavení A16 (P34 mikropočítače) do úrovně log. 0 a hodnoty v DPTR, kde DPH = 00H a DPL = adresa registru v RTC. Adresovat lze samozřejmě i registry R0 a R1 mikropočítače.

Čtení dat z obvodu. Data jsou čtena v součinnosti adresových signálů A0 až A3 a signálů CS2 a RD\. Mikropočítač vykoná obyčejnou funkci čtení svým signálem RD\. Číst lze instrukcí MOVX A,@DPTR, kdy se přesunou čtyřbitová data z příslušného registru do akumulátoru mikropočítače. Způsob vystavení adresy do DPTR je stejný, jako u zápisu dat do obvodu.

Udržení dat a chodu obvodu při výpadku a obnově napájení. Při výpadku napájení přejde výstup PF obvodu IO8 do úrovně log. 0 a tím i vstup CSO\ u RTC. Následuje pokles napájecího napětí na velikost napětí baterie. Toto napětí se musí pohybovat v rozsahu 2 až 5,5 V. Pokud napětí poklesne pod 2 V, obvod ztrácí data a vnitřní nastavení registrů. Při obnově napájení se nejprve zvětší napájecí napětí na 5 V a teprve potom přejde CS0\ do úrovně log. 1. Tímto způsobem je dodržen časový diagram, který je platný pro tento obvod.

Postup při prvním zápisu nebo při výměně záložní baterie. Je nutné nejprve nastavit řídicí registry, poté zapsat do obvodu datum, čas, rok a týden. Postupovat lze tak, že nejprve zapíšeme 0 do TEST a 1 do REST. Nastavíme mod čítání 12/24, potom REST do 0 a zastavíme činnost zápisem 1 do registru STOP. Nastavíme čas, datum, rok a týden a odblokujeme STOP nastavením do 0. Při zápisu je nutné číst registr BUSY a po zápisu nastavit HOLD do 0.

Postup při běžném zápisu. Čtením registru BUSY se informujeme o připravenosti obvodu k dalšímu zápisu. Při čtení BUSY musí být HOLD vždy ve stavu log.1.

Postup při čtení metodou HOLD. Nejprve zapíšeme 1 do HOLD a čteme BUSY. Pokud se nachází BUSY ve stavu 1, opakujeme čtení tak, že zapíšeme do HOLD 0 a poté 1. Pokud se BUSY nachází ve stavu 0, pokračujeme ve čtení a nakonec nastavíme HOLD do stavu 0.

Postup při čtení metodou DOUBLE. Při této metodě nevyužíváme čtení HOLD, avšak data čteme dvakrát. Pokud jsou data shodná, můžeme brát údaj za platný.

Postup při čtení vyvolaném externí žádostí o přerušení. Využíváme přerušení od obvodu RTC na jeho výstupu SDT s otevřeným kolektorem. Rezistor R12 zabezpečuje úroveň log. 1 při přítomnosti napájecího napětí 5 V. Po žádosti o přerušení je nutno vyčkat 36 μs ve dvanáctihodinovém cyklu nebo 3 μs ve dvacetičtyřhodinovém cyklu.

Hlídací obvod LTC691

Obvod IO8 (ADM691) monitoruje napájecí napětí a řídí zálohování napětí z baterie. V závislosti na stavu vstupů a na velikosti napájecího napětí vykonává RESET mikropočítače (nuluje ho), zálohuje napájecí napětí pro paměti, zakazuje zápis do pamětí a navíc obsahuje WATCHDOG.

Ve vnitřní struktuře obsahuje IO8 přesnou napěťovou referenci a komparátor, hlídající napájecí větev. Pokud je napětí mimo toleranci, aktivuje RESET a zakazuje výstupem CEO\ zápis do externích pamětí.

Napájecí napětí 5 V ze stabilizátoru IO9 je přivedeno na vstup Ucc. Zálohované napětí je potom přítomno na výstupu Uout. Baterie je připojena na vstup Ubat a je dobíjena přes rezistor R1 z výstupu Uout.

Při normálním provozu je interním tranzistorem NMOS sepnuto napájecí napětí Ucc na výstup Uout. Pokud je napájecí napětí menší než napětí na zálohovací baterii, je interně sepnuto napětí ze vstupu Ubat na výstup Uout.

Vstup PFI (power failrue input) je neinvertovaný vstup, který je vnitřně připojen na komparátor, jehož invertující vstup je připojen k vnitřnímu referenčnímu napětí 1,3 V. Jak je parné ze schématu, je vstup PFI připojen na rezistorový dělič R2, R3. Protože rezistor R2 je připojen za diodový můstek D2 až D5 před stabilizátorem IO9, je hlídáno přímo vstupní napětí. Kvůli rezistorům R2 a R3 a úbytku napětí na diodovém můstku je nutné na vstup X1 připojit napětí o minimální velikosti 9 V. Při poklesu napájecího napětí je aktivován výstup PFO (power failrue output), který je připojen na střed konektoru J6. Na tomto konektoru se zkratovací propojkou můžeme rozhodnout, zda bude u mikroprocesoru vykonáváno přerušení INT0 nebo INT1. Prostřednictvím přerušení se zjistí, zda pokleslo nebo vypadlo napájecí napětí. V podprogramu, který obsluhuje přerušení, můžeme potom vykonat funkce, které jsou potřebné ke zdárnému chodu programu po obnově napájení („uklidíme“ potřebná data do SRAM, EEPROM apod.). Je však nutné zajistit, aby přerušení od IO8 nekolidovalo s přerušením od RTC (IO10).

Výstup RST (RES) slouží k vykonání funkce RESET(nulování) u mikropočítače. Tato funkce je vyvolána při poklesu napájecího napětí pod úroveň 4,65 V nebo při neošetření WATCHDOGu. Při obnově napájení je automaticky generován nulovací signál o minimální délce 35 ms. Signál je přiveden i na výstupní konektory X2 a X4.

Vstup CE\I (chip enable input) není využit a je proto připojen na zem. Na tomto vstupu můžeme jinak nezávisle na vnitřních funkcích aktivovat výstup CE\O (chip enable output) přivedením úrovně log. 1.

Výstup CE\O je logický výstup, který v našem případě zakazuje zápis do pamětí. Signál je přiveden na dekodér adres IO6 a pokud je zápis zakázán kvůli poklesu napájecího napětí, způsobí svojí úrovní log. 1 neaktivnost dekodéru. Tento signál je taktéž vyveden na výstupní konektor.

Vstup WDI je nezapojen a v tomto stavu není funkce WATCHDOG uvolněna. WATCHDOG není od tohoto obvodu zapotřebí, protože jej obsahuje mikroprocesor.

Nyní popíšeme funkci IO8 v našem zapojení. Pro správnou funkci musí být připojena zálohovací baterie, jinak je obvod nečinný.

Při vypnutém napájení je na výstupu Uout napětí baterie, výstup PFO\ je v úrovni log. 0, výstup CE\O je v úrovni log. 1 a výstup RST je v úrovni log. 1.

Při zvyšování vstupního napětí *Ucc* nad velikost napětí baterie je na výstupu Uout napětí o velikosti napájecího napětí a baterie je dobíjena přes rezistor R1. Při dosažení vstupního napětí 4,7 V je odblokován RESET (výstup RST je v úrovni log. 0) a CE\O přejde do úrovně log. 0, výstup PFO\ však stále zůstává v úrovni log. 0. Po dosažení napětí 1,3 V na středu děliče R2, R3 (vstup PFI) přejde výstup PFO\ do úrovně log. 1 (v našem případě při napájecím napětí větším než 9 V).

Při výpadku napájení se mění stavové signály v opačném pořadí. Napětí díky vybíjení kondenzátorů v napájecí větvi plynule klesá. Nejprve přejde výstup PFO\ do úrovně log. 0 a pak CE\O a RST do úrovně log. 1.

Dekodér adres

Adresový prostor je rozdělen na dvě části, z nichž první část je adresována signály A0 až A15 a druhá signály A0 až A15 s využitím signálu P34/T0 jako adresového bitu A16. Tento adresový bit je nutno ovládat zvlášť příkazem assembleru - SETB T0 (nastavení do log.1) nebo CLR T0 (nastavení do log.0).

O vlastní rozdělení adresového prostoru se stará dekodér adres IO6 typu 74LS156 s výstupy typu otevřený kolektor. Dekodér je zapojen jako demultiplexer jedna z osmi. Na jeho vstupy jsou připojeny adresové bity A14 a A15 z mikropočítače, adresa A16 je vytvářena výstupem P34/T0.

Pokud jsou adresy A14 a A15 v úrovni log. 0 a adresový bit A16 je nataven do úrovně log. 1, je výstup 1Y0 v úrovni log. 0 a ostatní v úrovních log. 1. Přejde-li A14 do stavu log. 1, je výstup 1Y1 v úrovni log. 0 a ostatní v úrovních log. 1. Protože jsou výstupy 1Y0 a 1Y1 spojené, je umožněno od adresy 10000H až do adresy 17FFFH pracovat s první pamětí IO4, protože je signál CS0 je v úrovni log. 0.

Obdobně lze adresovat druhou paměť IO5 od adresy 18000H do 1FFFFH díky spojeným výstupům 1Y2 a 1Y3 dekodéru IO6 (je aktivní signál CS1).

Druhá část adresového prostoru je v součinnosti s A14 a A15 ovládaná adresovým bitem A16, který musí být nastaven do stavu log. 0. Od adresy 00000H do 03FFFH je aktivní CS2 (tj. CS2 je ve stavu log. 0), avšak pouze část od 00000H do 0000FH je využita pro spolupráci s RTC. Ostatní adresy tohoto prostoru od 00010H do 03FFFH nejsou interně využity.

Další vyšší adresy jsou využity pro aktivaci signálů CS3 až CS5, které jsou určeny pro ovládání externích obvodů (zapojených mimo desku). Signály CS3 až CS5 jsou přivedeny na výstupní konektory X3 a X5.

Vstupy 1G a 2G dekodéru IO6 jsou připojené k výstupu CE-O obvodu IO8. Signál CE\O způsobí neaktivnost CS0 až CS5 v případě, že napájecí napětí pokleslo pod minimální možnou úroveň, kdy nejsou obvody již schopné pracovat.

Všechny výstupy adresového dekodéru IO6 jsou otevřené kolektory tranzistorů. Rezistory R4, R5 a R7 zabezpečují úrovně log. 1 na vstupech CE\ pamětí a RTC (IO10) a jsou zapojeny na zálohované napětí *VCUB*. Úrovně log. 1 pro signály CS3 až CS5 jsou zabezpečovány rezistory R6, R8 a R9. Tyto rezistory jsou připojené ke sběrnici napájecího napětí 5 V bez zálohování a v případě výpadku napájení nemohou být aktivní.

V tab. 3 je přesné rozdělení celého paměťového prostoru dat.

Znovu upozorňujeme, a z tab. 3 je to patrné, že část adresového prostoru mezi nejvyšší

Tab. 3. Rozdělení paměťového prostoru dat

Adresa	Signál	Stav P34	Ovládaný prvek
10000H až 17FFFH	CS0	1	RAM (EEPROM) IO4
18000H až 1FFFFH	CS1	1	RAM (EEPROM) IO5
00000H až 0000FH	CS2	0	RTC
00010H až 03FFFH	CS2	0	volný prostor
04000H až 07FFFH	CS3	0	externí ovládání
08000H až 0BFFFH	CS4	0	externí ovládání
0C000H až 0FFFFH	CS5	0	externí ovládání

adresou pro spolupráci s RTC (0000FH) a nejnižší adresou pro aktivaci CS3 (04000H) je volný prostor. Protože však dekodér adres IO6 neumožňuje jemnější dělení, bylo by nutné na externí desce umístit zvláštní dekodér adres, který by tento paměťový prostor dokázal využít.

Sériový styk RS232

Pro obsluhu sériového styku RS232 je použit obvod IO7 typu ICL232. Obsahuje čtyři převodníky napěťových úrovní (dva pro výstup a dva pro vstup) a měnič napětí z 5 V na +10 V a -10 V. Kondenzátory C15 až C18 jsou filtrační.

Signál TXD (vysílání dat na portu P31 mikropočítače) o napěťové úrovni 0 V nebo 5 V, odpovídající logickým úrovním log. 0 a log. 1, je připojen na vstup IN1 a je vysílán obvodem IO7 na výstupu TD s napěťovými úrovněmi +10 V a -10 V. Druhý vysílaný signál je přiveden z P45 mikroprocesoru na vstup IN2 a je vysílán na výstupu RTS. Vstupní signály RD (přijímaná data) a CTS o úrovních +10 V a -10 V jsou přivedeny na vstupy RD a CTS obvodu IO7, který je převádí na úrovně 0 V a 5 V (výstupy OUT1 a OUT2), pak jsou zpracovávány na portech P30 a P35 mikropočítače. Propojka J8 určuje, zda port P35 bude využit pro zpracování signálu CTS nebo jako externí testovací vstup T1. Pokud nebude signál CTS využíván, je nutné ponechat propojku J8 nezkratovanou.

Sériové rozhraní RS232 umožňuje přenos signálů do vzdálenosti několika metrů. K mikropočítači lze samozřejmě připojit i jiné obvody sériového styku, např. RS485, který je schopen pracovat na větší vzdálenosti (až do 1 km). V tomto případě je nutné vyrobit externí desku zvoleného rozhraní a obvod IO7 neosadit, jinak by mohla nastat kolize signálů na portu P3 mikropočítače.

Napájení

Zařízení lze napájet ze zdroje stejnosměrného napětí 9 až 15 V nebo ze zdroje střídavého napětí v rozsahu 9 až 12 V. Napájecí napětí se připojí na konektor X1.

Při napájení stejnosměrným napětím nezáleží na polaritě, protože vstup je oddělen diodovým můstkem D2 až D5. Pokud chceme zachovat chod celého zařízení i při výpadku napájení, je vhodné zařízení napájet zálohovaným stejnosměrným zdrojem 13,8 V s baterií. Jinak jsou pouze zálohována data v SRAM a RTC interní baterií BT1.

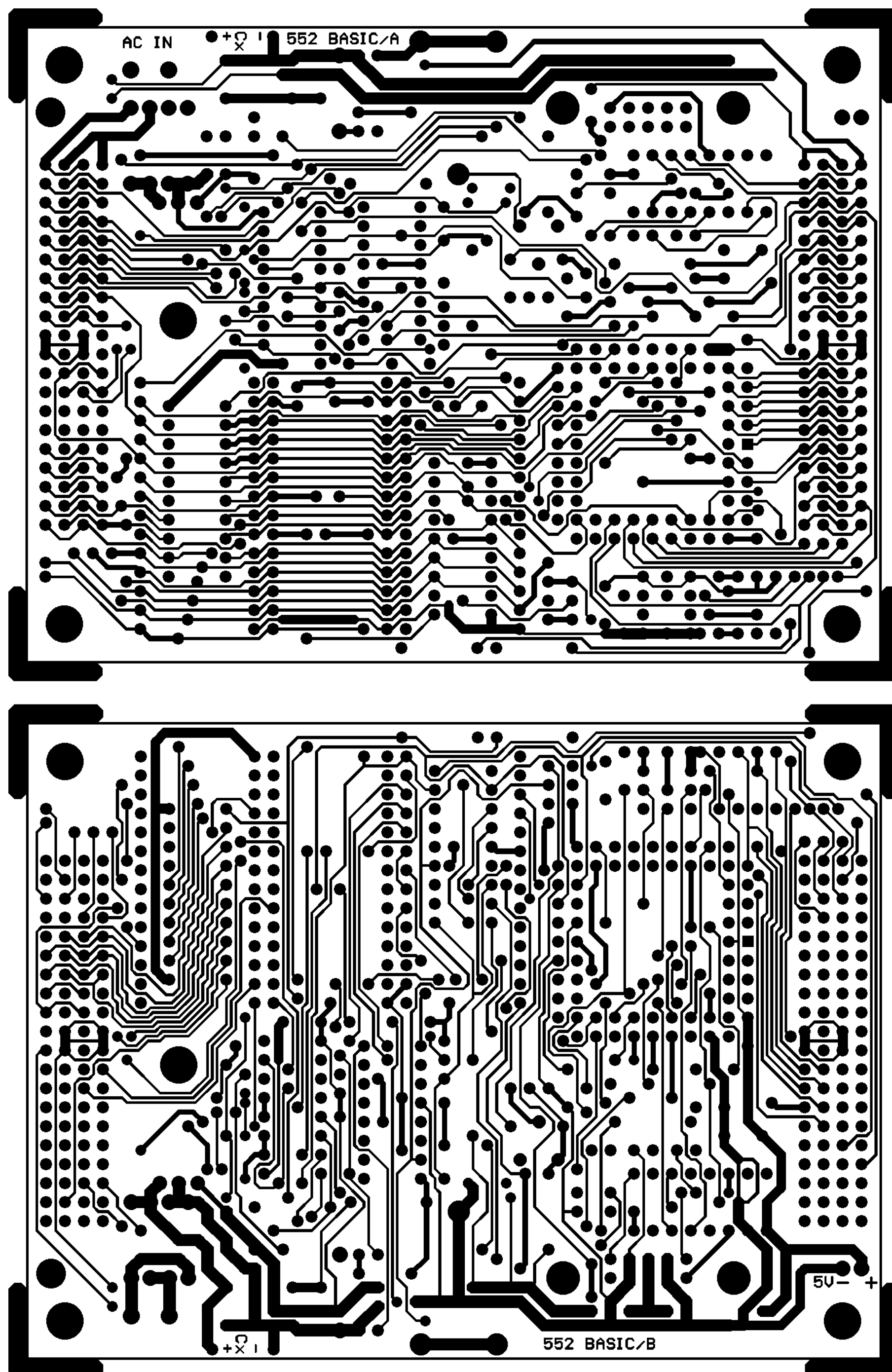
Na výstupní konektory je vyvedeno vstupní napětí ze zdroje a stabilizované napětí 5 V z výstupu IO9.

Pokud použijeme k napájení střídavé napětí, je vhodné posílit filtrační kondenzátory C12 a C13 vnějším přídavným kondenzátorem, který se připojí na pozici označenou symbolem Cx. Tato pozice se nachází na pravé straně základní desky vpravo dole a má označenou polaritu na osazovacím plánu. Připojením vnějšího filtračního kondenzátoru se prodlouží doba, během níž se mohou uklidit potřebná data při výpadku napájení po vyvolání přerušení hlídacím obvodem IO8. Filtrační kondenzátory C12 a C13 mají malou kapacitu a takové rozměry, aby bylo možné nad základní desku umístit další přídavnou desku (např. klávesnici s displejem) jednoduše nasunutím na výstupní konektory.

Přívod napájení by měl být chráněn tavnou pojistkou. U transformátorů napájecích zdrojů je nutné zajistit ochranu před přepětím v síti varistorem na vstupní straně 230 V. Větší přepětí v síti by mohlo poškodit obvody na desce a zničit drahé součástky.

Výstupní konektory

Výstupní konektory X2 až X5 jsou určeny ke spojení desky s dalšími obvody. Protože konektory jsou osazeny z obou stran desky



Obr. 2. Obrazec plošných spojů na straně A - straně pájení (nahoře) a na straně B - straně součástek (dole) základní desky BASIC-552 (měř.: 1 : 1)

s plošnými spoji, lze přídavné desky připojit tzv. „sendvičově" nad nebo pod základní desku BASIC-552.

Na konektory jsou vyvedeny všechny potřebné signály - adresová a datová sběrnice, porty P1, P3, P4, P5, vstupní napětí a stabilizované napětí 5 V, záložní napětí VCUB, signály ALE a PSEN\, výstupy pro pulzní řízení PWM0 a PWM1, signál RST, signál PF oznamující výpadek napájení a signály CS z dekodéru adres. Jediný signál, který není na konektory vyveden, je STADC, spolupracující s převodníkem AD. Pokud chceme tento signál využívat, je nutné propojit desky konektorem na pozici J5.

Na přídavné desky je vhodné umístit usměrňovací můstky a stabilizátory, které se napájejí vstupním napětím z vývodů 39 a 40 konektorů X3 a X5.

Veškeré vstupní a výstupní signály je vhodné oddělit od sběrnic obvody typu 74HCT245. Tím se zajistí funkce mikropočítače v případě, kdy se poškodí obvody na přídavných deskách.

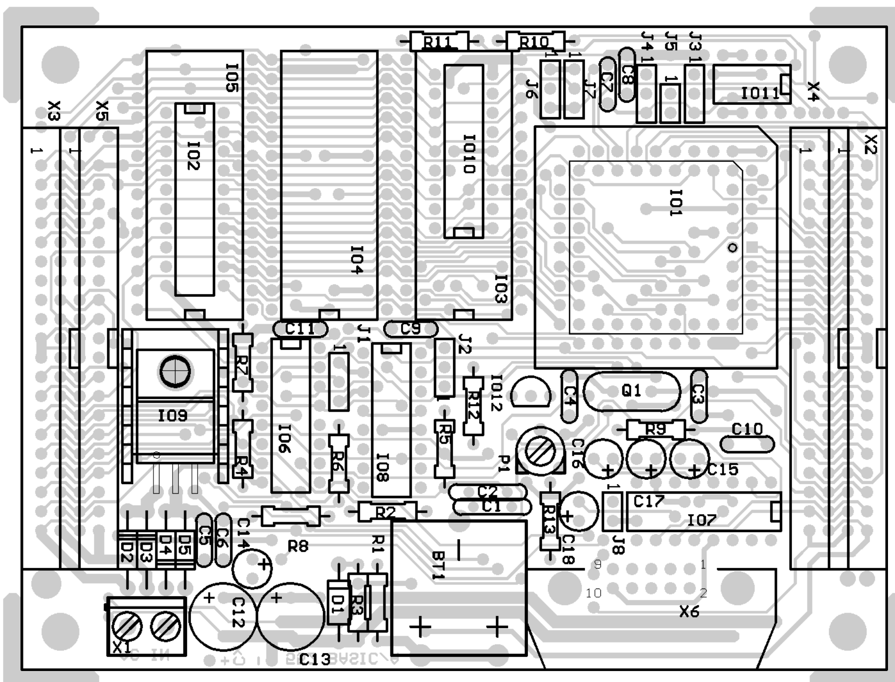
Výstupní signály z mikropočítače, které budou přímo řídit nějaké výstupy, musí být odděleny optočleny. Signály, připojené na port analogových převodníků P5, je vhodné elektricky oddělit převodníky DC/DC s dobrou linearitou.

Sběrnici zálohovaného napětí VCUB není vhodné příliš zatěžovat, protože při výpadku napájení je veškerá energie dodávána z baterie BT1. I když je baterie za provozu dobíjena, její kapacita je pouze 65 mAh.

Konektor X6 je vstupně/výstupní konektor sériového přenosu dat RS232. Pokud chceme použít jiný typ přenosu (např. RS485), obvody rozhraní je nutné umístit na přídavnou desku a využít signály RXD a TXD (P30 a P31).

Význam propojek

J1 - napájení paměti IO4. Při použití SRAM spojit střední kontakt s VCUB, při použití EEPROM spojit střední kontakt s Vcc.



Obr. 3. Rozmístění součástek na základní desce BASIC-552

J2 - napájení paměti IO5. Při použití SRAM spojit střední kontakt s VCUB, při použití EEPROM spojit střední kontakt s Vcc.

J3 - vstup EA\ mikropočítače. Při použití typu 80C552 spojit střední kontakt se zemí, při použití typu 87C552 spojit střední kontakt s Vcc.

J4 - vstup EW\ mikropočítače. Pro zákaz WATCHDOGu spojit střední kontakt s Vcc, pro uvolnění WATCHDOGu spojit střední kontakt se zemí.

J5 - vstup STADC mikropočítače. Spojit se zemí. Tento vstup nesmí být odpojený („ve vzduchu“).

J6 - vyvolání externího přerušení při poklesu napájecího napětí od IO8 (INT0 nebo INT1).

J7 - vyvolání externí žádosti o přerušení od RTC (INT0 nebo INT1).

J8 - funkce portu P35 mikropočítače. Má-li mikropočítač zpracovávat signál CTS z RS232, propojku spojíme, jinak ji necháme rozpojenou.

Stavba

Na obr. 2 jsou obrazce plošných spojů na straně pájení a na straně součástek, na obr. 3 je rozmístění součástek na desce. Deska je s oboustrannými spoji a s prokovenými otvory.

V amatérské praxi doporučujeme osazovat a oživovat desku po částech. Desku osazujeme podle dále uvedených bodů 1 až 6, zapojené dílčí části postupně oživujeme podle bodů 1 až 6 v kapitole Oživení.

1. Zdrojová část. Osadíme konektor X1, D2 až D5, C12 až C14, C5 a C6, R2 a R3 a D1.

Stabilizátor IO9 osadíme tak, že mezi obvod a desku s plošnými spoji vložíme chladič a dvě slídové podložky, a celek přišroubujeme k desce šroubem M3 s maticí a pérovou podložkou. Slídové podložky jsou dvě na sobě a musí být vloženy mezi desku a chladič - zabraňují zkratům spojů na desce s chladičem. Chladič musí být umístěn co nejblíže k vývodům stabilizátoru (stlačen dolů), aby bylo možné vložit paměť IO5 do objímky a pouzdro paměti se nedotýkalo chladiče. Šroub je vložen ze strany pájení. Před montáží potřeme chladič a křidélko stabilizátoru silikonovou vazelínou. Zkontrolujeme, zda je slídová podložka položena hranami rovnoběžně s chladičem. Vyčnívající závit šroubu zakápneme barvou (např. lakem na nehty).

2. Hlídací obvod napájecího napětí. Osadíme rezistor R1, kondenzátory C1 a C2 a integrovaný obvod IO8.

3. Zdroj referenčního napětí. Osadíme rezistor R13, trimr P1, kondenzátor C8 a integrovaný obvod IO12.

4. Integrované obvody. Osadíme IO2, IO6, obvod sériového styku RS232 IO7 a, pokud bude použit, také RTC IO10 (není dodáván se stavebnicí). Dále osadíme R4 až R9 a C15 až C18. Konektor X6 osadíme tak, že jej nejprve připevníme k desce s plošnými spoji dvěma šrouby M2,5. Šrouby vložíme ze strany pájení. Potom teprve vývody konektoru připájíme.

5. Osazení objímek pro IO. Nejprve osadíme objímku PLCC pro mikropočítač. Pozor, vzhledem k množství vývodů musíme objímku osadit se správnou orientací, jinak prakticky není možnost ji vypájet v amatérských podmínkách! Jedinou možností je nahřát horkovzdušnou pistolí vývody objímky na desce ze strany pájení tak, až objímka „vypadne“, a potom odsát cín z děr odsávačkou. Objímka je orientována tak, že její zkosený roh je vlevo nahoře (jak je patrné z osazovacího plánu na obr. 3). Před pájením je nutné objímku přitisknout na desku.

Objímky pro IO3 a IO5 je nutné mechanicky upravit. Z obou odstraníme štípačkami středové rozpěrky a odstraníme otřepy. U objímky pro IO3 je ještě nutné zapilovat spodní rozpěrku. Pokud nebudeme používat obvod IO10, je vhodné jej (nebo jiný obvod ve stejném pouzdru) zkusmo vložit do desky, potom objímku pro IO3 upravit, zapájet a obvod vyjmout. Osadíme objímky pro IO4 a IO11 a rezistory R10 až R12.

6. Osazení zbylých součástek. Osadíme C3 a C4, C7, C9 až C11, kontaktní kolíky pro propojky J1 až J8 (které vyrobíme rozlomením delší kontaktní lišty S1G40 na příslušné díly) a konektory X2 až X5.

Konektory X2 a X3 jsou umístěné na straně pájení a konektory X4 a X5 jsou umístěné na straně součástek. Konektor X5 je nutné osadit tak, aby jeho hrany lícovaly s hranami konektoru X3 (nesmí být osazen o jednu pozici vpravo, jak to umožňují díry na desce)!

Nakonec osadíme krystal tak, že jej podložíme izolační fólií nebo papírem a zapájíme ho. Při použití mikropočítače 80C552 a paměti SRAM typu 62256 umístíme zkratovací propojky tak (při pohledu na desku, kdy konektor X6 je vpravo nahoře a mikropočítač je v levém horním rohu), že na kontaktních kolících J1 je zkratovací propojka umístěná vpravo, na J2 vlevo, na J3 vlevo, na J4 vpravo, kontaktní kolíky J5 jsou propojené.

Ostatní propojky mohou zatím zůstat volné.

Oživení

Oživení je poněkud náročnější proces, který vyžaduje určité zkušenosti. K základnímu oživení je zapotřebí ohmmetr, regulovatelný zdroj ss napětí a osciloskop. Při oživování postupujeme podle následujících bodů.

1. Oživení zdrojové části. Na konektor X1 přivedeme stejnosměrné napětí 12 V ze zdroje a zkontrolujeme napětí 5 V na výstupu stabilizátoru IO9. Taktéž je vhodné zkontrolovat přítomnost napětí 5 V na příslušných napájecích ploškách u všech integrovaných obvodů.

2. Oživení hlídacího obvodu napájecího napětí. Pokud osazujeme desku po částech, připojíme baterii BT1 (dodržujeme připojení se správnou polaritou). Baterii není vhodné zatím pájet do desky, připojíme jí externě pomocí vodičů. Přivedeme napájecí napětí 12 V ze zdroje. V tomto stavu musí být na vývodech IO8 následující úrovně napětí: na Uout (vývod *2*) je napětí 5 V, na PFO\ (vývod *10*) je úroveň log. 1, na CE\O (vývod *12*) je úroveň log. 0 a na RST (vývod *16*) je úroveň log. 0.

Po změření těchto úrovní připojíme voltmetr na výstup PFO\ (vývod *10*) a postupně snižujeme napájecí napětí tak, až se na tomto výstupu objeví úroveň log. 0. Překlopení do úrovně log. 0 musí nastat při velikosti vstupního napájecího napětí okolo 8,5 až 9 V. Potom dále zmenšujeme napájecí napětí a kontrolujeme stav na výstupech RST a CE\O. Při jejich překlopení do stavu log. 1 změříme napětí na výstupu stabilizátoru, které musí být v rozmezí 4,6 až 4,8 V. Tak jsme zkontrolovali funkčnost obvodu IO8, a pokud osazujeme desku po částech, odpojíme napájecí napětí a baterii a pokračujeme v osazování.

3. Oživení zdroje referenčního napětí. Přivedeme napájecí napětí a zkontrolujeme přítomnost referenčního napětí na „katodě“ IO12 a na vývodu *59* objímky mikroprocesoru IO1. Trimr P1 natočíme na levý doraz a znovu změříme referenční napětí, totéž měření provedeme při natočeném trimru na pravý doraz. Napětí se musí pohybovat v minimálním rozsahu 2,35 až 2,6 V.

4. Kontrola napájecího napětí na vývodech IO6 a IO10. Po připojení vstupního napájecího napětí zkontrolujeme napájecí napětí přímo na vývodech IO6 a IO10.

5. Kontrola napájecího napětí na vývodech IO1, IO3 až IO5 a IO11.

6. Celkové oživení. Než vložíme mikropočítač, zkontrolujeme napětí na vývodech jeho objímky. Upozorňujeme, že vývod *1* u mikropočítače není, jako u standardních součástek, u zkoseného rohu, ale je umístěn uprostřed horní strany. Další vývody jsou rozmístěné postupně vlevo proti směru hodinových ručiček.

Připojíme baterii a napájecí napětí 12 V. Na jednotlivých vývodech musí být následující napětí: na vývodu *2* (Ucc) napětí 5 V, na vývodu *3* napětí 0 V, na vývodu *6* napětí 5 V, na vývodu *15* napětí 0 V, na vývodu *36* a *37* napětí 0 V, na vývodu *49* napětí 0 V, na vývodu *58* napětí 0 V, na vývodu *59* napětí *Uref*, na vývodu *60* napětí 0 V a na vývodu *61* napětí 5 V.

Odpojíme napájecí napětí a vložíme mikropočítač a nenaprogramovanou paměť EPROM (s daty 0FFH) 27C512 na pozici IO3. Vhodnější je celou paměť naprogramovat daty 00FH, při

kterých mikropočítač vykonává funkci NOP (žádná operace).

Znovu připojíme napájecího napětí a zkontrolujeme osciloskopem, zda jsou generovány signály ALE a PSEN\. Zkontrolujeme, zda mikropočítač generuje signály na datové sběrnici a adresy na výstupech IO2 a na svých portech P20 až P27. Měřením na portech P10 až P17, P30 až P37 a P40 až P47 musí být zjištěno, že jsou všude vystaveny úrovně log. 1.

Je vhodné zkontrolovat, že odpovídající si datové a adresové signály na vývodech paměti EPROM a na kontaktech objímek IO4 a IO5 jsou shodné (to však neznamená, že jsou shodné signály na stejných vývodech. Tak např. adresový signál A14 je u EPROM na vývodu *27* a na objímkách IO4 a IO5 na kontaktech *1*).

Na vývodech *22* a *27* (OE\ a WE\) IO4 a IO5 musí být stav log. 1.

Pokud jsou porty P31 a P45 ve stavu log. 1, musí být na výstupech TD a RTS obvodu IO7 záporné napětí.

Trimrem P1 nastavíme referenční napětí na katodě IO12 na potřebnou velikost.

Před připájením baterie zkontrolujeme odběr z baterie v klidovém stavu tak, že jí připojíme k desce v sérii s ampérmetrem, přičemž odběr nesmí být větší než několik µA. Baterii pájíme při odpojeném napájecím napětí. Po připojení napájecího napětí se musí napětí baterie zvětšit díky jejímu dobíjení - měříme voltmetrem.

Seznam součástek

R1	1,5 kΩ, miniaturní	
R2	15 kΩ, miniaturní	
R3	3,3 kΩ, miniaturní	
R4, R5, R6, R7, R8, R9, R10, R11	12 kΩ, miniaturní	
R12	100 kΩ, miniaturní	
R13	2,2 kΩ, miniaturní	
P1	10 kΩ, trimr PT6V	
C1, C2, C5, C6, C7, C8, C9, C10, C11	100 nF, keramický	
C3, C4	27 pF, keramický	
C12, C13	220 µF/25 V, rad.	
C14 až C18	22 µF/25 V, rad.	
D1	BZX85V004.7	
D2 až D5	1N4002	
IO1	80C552	
IO2	74HCT573	
IO3	27C512	
IO4, IO5	28C256 nebo 62256	
IO6	74LS156	
IO7	ICL232	
IO8	ADM691	
IO9	7805	
IO10	RTC72421	
IO11	93C56	
IO12	LM336	
Q1	krystal 11,0592 MHz	
X1	ARK500/2	
X2 až X5	BL840GD	
X6	PSL10W	
BT1	B-Z3A65LF2	
kontaktní lišta S1G40		1 kus
objímka PLCC68Z		1 kus
objímka DIL28PZ		3 kusy
objímka DIL8PZ		1 kus
zkratovací propojka JUMP-RT		8 kusů
chladič DO2A		1 kus
slídová izolační podložka TO220		2 kusy
deska s plošnými spoji 552 BASIC		

využité), zapsáním stavu log. 0 na příslušnou pozici se LED rozsvítí.

LED jsou umístěné na desce tak, aby jejich středy byly ve středu os řádků klávesnice.

Akustická signalizace

Akustickou signalizaci BEEP („pípnutí") umožňuje obvod IO7 typu NE556, což je zdvojený časovač 555. IO7A pracuje v monostabilním režimu a na sestupnou hranou impulsu na vstupu TR vygeneruje kladný impuls na výstupu Q. Délka tohoto impulsu určuje trvání akustického signálu. Rezistorem R1 a kondenzátorem C10 je nastavena časová konstanta.

Druhý obvod IO7B pracuje v astabilním režimu a kmitočet výstupního signálu určuje výšku tónu. Časová konstanta obvodu je určena rezistory R3 a R4 a kondenzátorem C1. V klidovém stavu je obvod nulován výstupem z IO7A.

Signál z výstupu Q IO7B je přiveden přes rezistor R5 na bázi tranzistoru T1, kterým je buzen reproduktor RE1. Během vývoje byl místo reproduktoru vyzkoušen piezoměnič (můstkově buzený), při napájení 5 V s ním však nebylo dosaženo potřebného akustického tlaku (signál byl slabý).

Paralelní kombinace rezistorů R6 a R7 omezuje proud do reproduktoru. Akustická signalizace je aktivována signály A11, CS3 a WRN přes obvody IO5D a IO9D na adrese 04800H a na všech dalších adresách, u nichž je adresový bit A11 ve stavu log. 1 (vzhledem k dekodéru na desce BASIC-552).

R2 a C2 nulují obvod IO7A při připojení napájecího napětí.

Dekodér adres

Aby bylo možné spolupracovat s klávesnicí a displejem, je nutné zajistit přístup k nim na určitých adresách. O to se stará dekodér adres IO2 s okolními obvody. Je nutné si uvědomit, že dekódování adres je již vztaženo k dekodéru na desce BASIC-552, který řídí signál CS3, takže adresování může probíhat v rozsahu 06000H až 06007H vzhledem k zapojení vstupů G dekodéru na výstup obvodu IO4C, který je ovládán negovaným signálem CS3 a adresovým bitem A13.

Na adrese 06000H je výstup dekodéru 2Y0 ve stavu log. 0, ostatní výstupy jsou ve stavech log. 1. Na dalších adresách 06001H až 06003H jsou postupně ve stavu log. 0 výstupy 2Y0 až 2Y3. Všemi čtyřmi výstupy 2Y je řízeno čtení z klávesnice.

Na adresách 06004H až 06007H jsou postupně ve stavech log. 0 výstupy 1Y0 až 1Y3, jimiž je ovládán displej LCD (zápis a čtení dat).

Zapojení displeje

Komunikace s displejem nezávisí pouze na vhodném adresování, je také nutné aktivovat jeho řídicí signály E (enable), R/W (read/write) a RS (register select). S displejem je možné komunikovat, pokud je na vstupu E úroveň log. 1, což je vždy při příchodu signálů RDN nebo WRN na příslušné adrese od mikroprocesoru.

Na adrese 06004H jsou vstupy displeje R/W a RS ve stavu log. 0 a je umožněn zápis řídicí instrukce do displeje. Na adrese 06005H je vstup R/W ve stavu log. 0 a RS ve stavu log. 1, čímž je umožněn zápis dat do displeje. Na adrese 06006H je vstup R/W ve stavu log. 1 a RS ve stavu log. 0 a je možné číst BUSY flag a adresový čítač displeje. Konečně na adrese 06007H jsou oba vstupy R/W a RS ve stavu log. 1 a je možné číst data z CG RAM nebo z DD RAM displeje.

Obousměrná datová sběrnice displeje D0 až D7 je připojena přes oddělovací obvod IO1 k datové sběrnici desky BASIC 552.

KEYDSP1 - deska klávesnice a displeje

Základní deska mikropočítače BASIC-552 může vykonávat samostatné funkce, ve většině případů je však požadováno zadávat určitá data prostřednictvím klávesnice a zobrazovat údaje na displeji, což umožňuje tato deska. Kromě klávesnice a displeje obsahuje deska ještě čtyři samostatně ovládané LED a obvod pro akustickou signalizaci (BEEP).

Je nutné upozornit, že veškeré data a adresy jsou vztažené k signálům, které je schopna generovat deska BASIC-552. Adresový bit A16 je na kartě BASIC generován portem P34 (T0) a je ho nutno ovládat softwarově. Pro ovládání klávesnice a displeje musí být tento bit ve stavu log. 0. Bit A16 lze nastavit instrukcí CLR P3.4 assembleru pro mikropočítače řady „51".

Deska je s oboustrannými plošnými spoji s prokovenými otvory. Ze strany pájení jsou umístěné konektory X2 a X3 a reproduktor. Umístění konektorů X2 a X3 na desce a zapojení jejich vývodů je přizpůsobeno ke spojení karty se základní deskou BASIC-552. Protože je na desce na straně součástek umístěn displej LCD a klávesnice, není samozřejmě možné stavebnici rozšiřovat nad tuto desku (deska ani neobsahuje přípojné konektory na straně součástek). Všechny další desky stavebnice je nutné umístit pod základní desku BASIC-552 nebo je nutné je připojit plochými kabely.

Popis zapojení

Schéma desky KEYDSP1 je na obr. 4. Deska obsahuje oddělovací obvod dat IO1 (74HCT245), inteligentní dvouřádkový LCD displej se šestnácti znaky na každém řádku, klávesnici se šestnácti tlačítky, čtyři LED s malým příkonem, obvody pro BEEP s reproduktorem, dekodér adres a zdroj.

Obvody desky jsou napájené z vývodů *39* a *40* (ST1 a ST2) konektoru X3. Potřebné ovládací signály jsou: datová sběrnice D0 až D7, adresová sběrnice se signály A0 až A2 a A11 až A14, signál CS3 z dekodéru adres a signály RD\ a WR\.

Datová sběrnice

Datová sběrnice je oddělena obvody IO1 a IO6. Obvod IO1 je obousměrný budič sběrnice a slouží k oddělení datových signálů mezi deskou BASIC-552 a displejem DSP1. Vstup G obvodu IO1 umožňuje uvést výstupy obvodu do třetího stavu, vstup DIR řídí směr toku dat. Tím je umožněno jednak do displeje zapisovat (instrukce a potřebná data) a zároveň z displeje data číst.

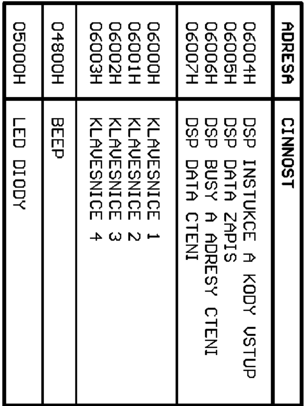
Obvod IO6 odděluje datovou sběrnici od matice tlačítek. Oba obvody jsou typu HCT (high CMOS s úrovněmi, slučitelnými s obvody TTL) kvůli malému zatěžování sběrnice. Způsob ovládání klávesnice a displeje bude popsán v následujících odstavcích.

LED

Čtyři LED D9 až D12 lze využít pro optickou signalizaci. Obvod IO10 je čtyřbitový záchytný registr a umožňuje ovládat LED datovými signály D0 až D3 v součinnosti se signály A12, CS3 a WRN.

Obvody IO5C, IO9C a IO11D tvoří adresový dekodér, který umožňuje zápis do obvodu IO10 na adrese 05000H a na všech dalších adresách, u nichž je adresový bit A12 v úrovni log. 1 (vzhledem k dekodéru na kartě BASIC 552 a využití signálu CS3).

Obvod IO10 je typu LS kvůli odběrům proudu na jeho výstupech. Do obvodu se zapisují pouze čtyři bity dat (D0 až D3 - ostatní jsou ne-



ADRESA	CINNOST
0604H	DSP INSTUKCE A KODY USTUP
0605H	DSP DATA ZAPIS
0606H	DSP BUSY A ADRESY CTENI
0607H	DSP DATA CTENI
0600H	KLAVESNICE 1
0601H	KLAVESNICE 2
0602H	KLAVESNICE 3
0603H	KLAVESNICE 4
4800H	BEEP
5000H	LED DIODY

KEYDSP1 12/99

Obr. 4. KEYDSP1 - deska klávesnice a displeje

Trimrem R8 se v rozsahu 0 až 5 V ovládá napětí na vstupu V0 displeje a tím se nastavuje jas zobrazovaných znaků. Zvětšováním napětí na V0 se sytost znaků zmenšuje. U použitého displeje je optimální sytost znaků při napětí asi 0,5 V.

Zapojení klávesnice

Po dekódování adres z IO2 (jak již bylo popsáno) je možné na adresách 06000H až 06004H číst stav klávesnice (nikoliv do ní zapisovat).

Vnitřní zapojení klávesnice je takové, že jsou navzájem elektricky spojené jednotlivé řádky a sloupce, čímž je vytvořena matice 4x 4. Aby nenastala kolize signálů na datové sběrnici, pokud by byla nechtěně vykonána instrukce zápisu, uvolní obvod IO11 přístup na sběrnici přes IO6 pouze v součinnosti se signálem RD\ (čtení).

Diody D5 až D8 oddělují jednotlivé sloupce matice a kvůli malým úbytkům napětí byly zvolené diody typu Schottky (BAT46). Rezistory R9 až R12 zajišťují úrovně log. 1 na řádcích matice. Obvod IO6 je třístavový čtyřnásobný budič sběrnice a je aktivní, když je na jeho spojené řídicí vstupy přivedena úroveň log. 1.

Střední osa klávesnice je umístěná v příčné ose displeje. Toto uspořádání umožňuje např. na dolní řádce displeje zobrazovat menu, ke kterému bude přináležet horní sada čtyř tlačítek.

Menu musí být samozřejmě zobrazováno formou zkratek, protože ke každému tlačítku přísluší pouze čtyři znaky, a to ještě bez mezer. Jednotlivé položky menu lze rozlišit např. odlišným písmem (malá a velká písmena) nebo blikáním znaků na příslušných pozicích.

Zdroj

V profesionální praxi se většinou nepoužívá jeden stabilizátor pro napájení všech připojených desek. Proto tato deska obsahuje samostatný stabilizátor, který je napájen z desky BASIC-552.

Na vstupu je diodový můstek s diodami D1 až D4, za ním filtrační kondenzátory C5 a C11 a stabilizátor IO8. Deska může být tudíž napájena střídavým nebo stejnosměrným napětím v rozsahu 9 až 15 V přes vývody konektoru X3.

Na výstupu stabilizátoru je filtrační kondenzátor C12 a na desce jsou rozmístěny v napájecí větvi další filtrační keramické kondenzátory o kapacitě 100 nF (C6 až C9).

Ovládání klávesnice, displeje, LED a BEEP

Čtení klávesnice

Instrukcí MOVX A,@DPTR assembleru 51 lze číst stav klávesnice tak, že adresa se předem uloží do registru DPTR a do akumulátoru jsou přenesena čtyřbitová data (D0 až D3) z výstupů IO6. Datové signály D4 až D7 jsou bezvýznamné a v programu je možné tyto bity zamaskovat. Při vlastní komunikaci jsou čtena postupně data na adresách 06000H až 06003H.

Proces čtení klávesnice lze popsat takto:

Čtením na adrese 06000H se výstup 2Y0 uvede do stavu log. 0 a přes diodu D8 je připojen první sloupec matice klávesnice k zemi. Pokud není stisknuté žádné tlačítko, je díky rezistorům R9 až R12 čten stav XFH. Stiskne-li se kterékoliv tlačítko v prvním sloupci, je na příslušné pozici čtena úroveň log. 0. Totéž se děje i na následujících adresách. Tímto způsobem lze úplně přečíst stav klávesnice, a to i tehdy, je-li stisknuto více tlačítek současně.

Protože tlačítka nejsou bezzákmitová, je nutné číst stav klávesnice několikrát opakovaně za sebou. Většinou se doporučuje číst stav klávesnice čtyřikrát až pětkrát za sebou ve dvacetimilisekundových intervalech, a pokud jsou čtená data shodná, vyhodnotit stav za platný. Stejným způsobem je vhodné vyhodnotit stav uvolnění klávesnice.

Pokud chceme vyhodnocovat stav dvojstisků na klávesnici, je nutné prodloužit interval mezi jednotlivými čteními na 50 ms nebo číst stav klávesnice minimálně 10x. Je to nutné z toho důvodu, že uživatel v podstatě nikdy nestiskne dvě tlačítka zcela současně a prodleva mezi jednotlivými stisknutími může být řádově až několik set milisekund.

Ovládání displeje

Displej je dvouřádkový a obsahuje 16 znaků na každém řádku, takže je možné zobrazit celkem 32 znaků.

Znakový formát každého pole znaku je 5x 7 bodů s kurzorem. Na displeji se znaky, které jsou uloženy ve znakovém generátoru ROM, mohou jednoduše zobrazit zapsáním dat, nebo je možno vytvořit osm různých uživatelsky definovaných znaků. Na displeji je tedy možno zobrazovat alfanumerické znaky a různé symboly. Displej přijímá instrukce, data a adresy z mikropočítače, data uchovává a dekóduje je na matici 5x 7 bodů znaku. Displej obsahuje CG RAM = znakový generátor RAM a DD RAM = datovou znakovou paměť RAM.

Znakový kód je formou dat zapsán do datového registru. Postup programování lze popsat takto:

Inicializace displeje. Protože displej obsahuje řadič HD44780 nebo ekvivalentní a tento typ řadiče je použit ve více druzích displejů, je nutné provést inicializaci pro daný typ.

1. Po zapnutí napájecího napětí je nutné vyčkat déle než 15 ms od okamžiku, kdy napájecí napětí překročí 4,5 V.

2. Protože displej je schopen pracovat se čtyřbitovými daty, a v našem případě posíláme a čteme data osmibitová, je nutné nejprve provést instrukci Function Set (nastavení funkce) zápisem 001110xxB na adresu 06004H. Tím je zároveň displej inicializován jako dvouřádkový.

Nyní je vhodné číst BF (flag BUSY) na adrese 06006H bit D7 nebo vyčkat déle než 4,1 ms na dokončení vnitřní činnosti v displeji. Pokud BF nebude čten, je nutné zapsat instrukci Function Set ještě třikrát (nevím proč, je to však předepsáno v inicializačním diagramu. V praxi jsem zápis provedl pouze jednou, četl jsem BUSY a displej pracoval normálně).

3. Zápisem dat 00001000B na adresu 06004H se vloží instrukce Display off (vypnutý - nezobrazuje znaky).

4. Po přečtení BUSY zápisem 00000001B na adresu 06004H je vymazáno pole displeje, displej je připraven na funkci automatického posunu znaků a je vynulován adresový čítač DD RAM.

5. Opět po přečtení BUSY zápisem dat 00000111B na adresu 06004H je nutné nastavit vstupní mód (Entry Mode Set).

V tomto případě je při jakékoliv operaci kurzor automaticky posouván vpravo (inkrement). Pokud budeme potřebovat posunovat kurzor vlevo, je nutné na pozici dat D1 zapsat 0 a pokud nechceme posouvat kurzor, zapíše se na pozici D0 stav 0.

Tímto způsobem je provedena základní inicializace a je možné zapisovat data pro výpis znaků.

Pokud bylo postupováno podle předcházejících instrukcí, kurzor je umístěn v horním řádku vlevo (na pozici HOME) a bliká.

Nulování displeje (clear display). Displej se vynuluje zápisem dat 00000001B na adresu 06004H. Při vynulování se nastaví automaticky adresový čítač DD RAM na adresu 0 a kurzor je umístěn v levém horním rohu.

Návrat kurzoru (return home). Adresový čítač DD RAM se vynuluje zápisem dat 00000010B na adresu 06004H. Obsah DD RAM zůstává přitom nezměněn.

Nastavení vstupního módu (Entry Mode Set). Tímto modem se určuje funkce kurzoru. Vstupní mod se nastaví zápisem dat 000001xyB na adresu 06004H. Bit x určuje, zda adresa v čítači pro DD RAM se bude zvyšovat nebo snižovat o jednotku. Pokud je x = 1, adresa je inkrementována, pokud je x = 0, adresa je dekrementována. Bit y určuje, zda kurzor bude automaticky přesouván při každém zápisu nebo čtení dat (y = 1 znamená, že posouvání je zapnuto).

Displej ON/OFF. Instrukce se aktivuje celý displej, kurzor a jeho blikání. Instrukce se vykoná zápisem dat 00001xyzB na adresu 06004H. Bit x = 1 znamená aktivaci displeje, bit y = 1 aktivuje kurzor a z = 1 uvolňuje blikání znaku na pozici kurzoru.

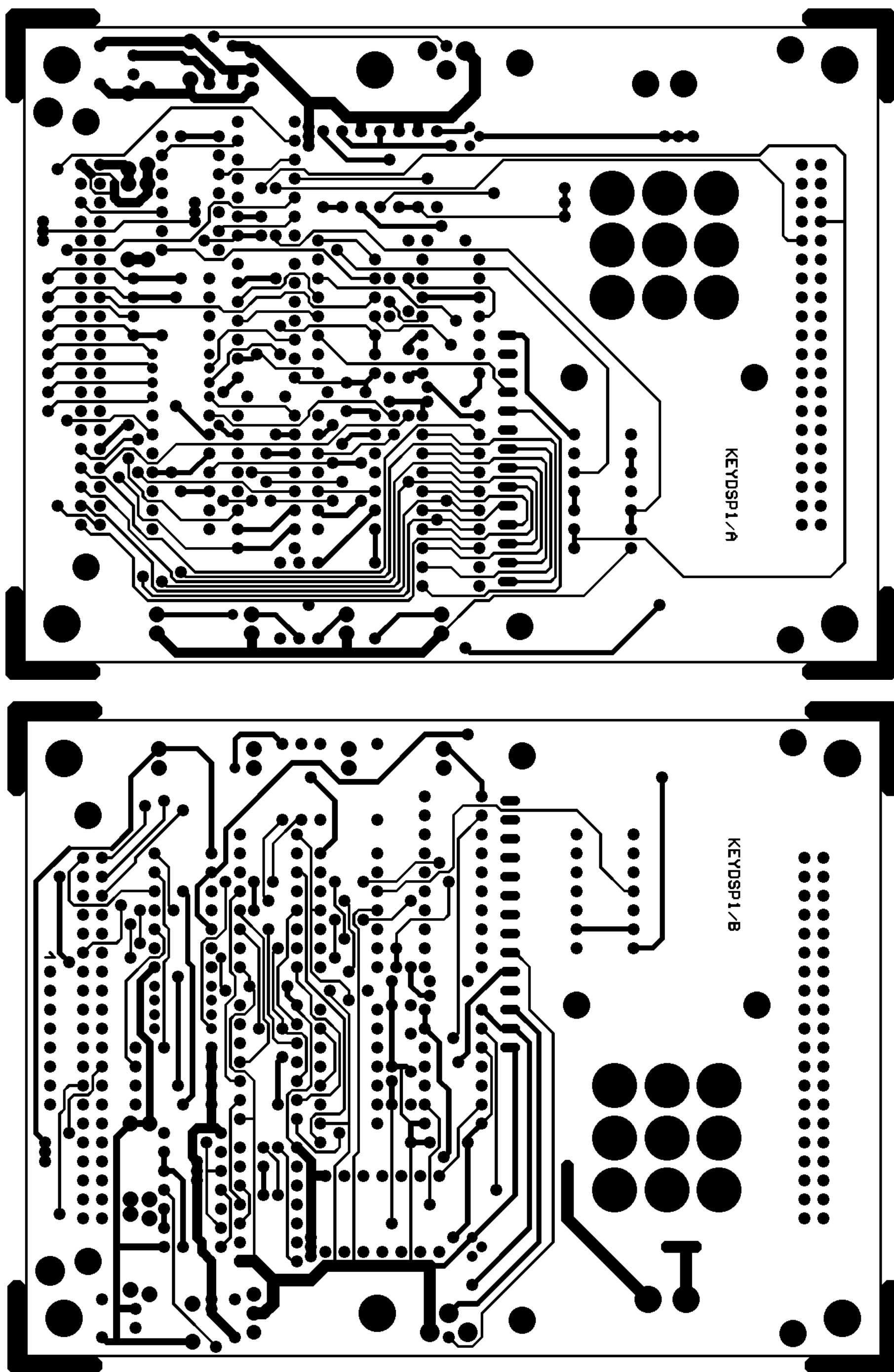
Přesun znaků/kurzoru (Display/Cursor Shift). Zápisem instrukce je nastaven mod přesouvání znaků nebo kurzoru. Instrukce se vykoná zápisem dat 0001xyzzB na adresu 06004H. Bit x = 1 uvolní posouvání znaků, x = 0 umožní přesun kurzoru. Bit y určuje, jakým směrem bude přesun probíhat (y = 1 uvolní přesun vpravo a y = 0 uvolní přesun vlevo od stávající pozice). Bit z může být v libovolném stavu.

Nastavení funkce (Function Set). Před vlastní inicializací je nutné určit, jaká délka datových slov bude přenášena (4 nebo 8 bitů) a kolik má displej řádků. Funkce se nastaví zapsáním dat 001xy0zz na adresu 06004H. Bit x určuje délku datového slova (v našem zapojení musí být vždy x = 1), bit y určuje počet řádků (v našem případě musí být opět y = 1). Bit z může být v libovolném stavu.

Nastavení adresy CG RAM displeje (CG RAM Addres Set). Adresa se nastaví zápisem dat 01AAAAAAB na adresu 06004H, přičemž A představuje jednotlivé bity adresy.

Tab. 4. Úplný přehled instrukcí displeje s řadičem HD44780

Instrukce	Činnost	Adresa	D7			- Data -				D0	Doba
nulování displeje	zápis	06004H	0	0	0	0	0	0	0	1	1,64 ms
kurzor na počátek	zápis	06004H	0	0	0	0	0	0	1	x	1,64 ms
nastavení vstup. modu	zápis	06004H	0	0	0	0	0	1	ID	S	40 μs
displej ON/OFF	zápis	06004H	0	0	0	0	1	D	C	B	40 μs
displej/kurzor přesun	zápis	06004H	0	0	0	1	SC	RL	x	x	40 μs
nastavení funkce	zápis	06004H	0	0	1	DL	N	0	x	x	40 μs
nast. adresy CG RAM	zápis	06004H	0	1	---------- Acg ---------------						40 μs
nast. adresy DD RAM	zápis	06004H	1	--------------- Add ---------------							40 μs
čtení BUSY a adres. čítače	čtení	06006H	BF	--------------- AC ----------------							0 μs
zápis dat	zápis	06005H	------------ zapisovaná data --------								40 μs
čtení dat	čtení	06007H	---------------- čtená data ------------								40 μs



Obr. 5. Obrazec plošných spojů na straně A - straně pájení (nahoře) a na straně B - straně součástek (dole) desky klávesnice a displeje KEYDSP1 (měř.: 1 : 1)

Nastavení adresy DD RAM displeje (DD RAM Addres Set). Adresa se nastaví zápisem dat 1AAAAAAAB na adresu 06004H, přičemž A představuje jednotlivé bity adresy.

Flag BUSY a čtení adresového čítače (Busy Flag/Addres Counter Read). Čtením flagu BUSY se informujeme, zda je displej zaměstnán vnitřními operacemi, nebo zda je schopen spolupracovat s okolím. Čtením adresového čítače je umožněno sledovat pozice znaků. Obě informace se čtou na adrese 06006H instrukcí MOVX A,@DPTR assembleru 51, kde adresa musí být předem uložena v registru DPTR. Flag BUSY je umístěn jako bit D7, ostatní bity připadají na adresu.

Zápis dat do displeje (CG RAM/ /DD RAM Data Write). Po zápisu osmibitových dat na příslušné adresy adresového čítače jsou tato data dekódována a příslušné znaky jsou vypsány na displej. Data se zapisují na adrese 06005A. Zápisem se může automaticky inkrementovat nebo dekrementovat stav adresového čítače, pokud je to umožněno přednastavením bitem D1 v instrukci nastavení funkce.

Čtení dat z displeje (CG RAM/DD RAM Data Read). Dat z CG RAM a DD RAM je možné číst na adrese 06007H instrukcí MOVX A,@DPTR assembleru 51, kde adresa musí být předem uložena v registru DPTR. Před vlastním čtením je nutno nastavit adresu do příslušné RAM. Jestliže nebude nastavena, první čtená data nebudou platná.

Úplný přehled instrukcí je uveden v tab. 4. Všimněte si, že řadič displeje čte postupně data od D7 do D0, a když narazí na bit ve stavu log. 1, zjistí z jeho umístění, o jakou instrukci se skutečně jedná. Samozřejmě vše se odvíjí od stavů hardwarových vstupů E, R/W a RS.

Dále je vysvětlen význam jednotlivých proměnných z tab. 4.

ID = 1 - inkrementace adresy kurzoru.
ID = 0 - dekrementace adresy kurzoru.
S = 1 - přesouvání displeje zapnuto.
D - aktivace displeje.
C - aktivace kurzoru.
B - blikání znaku na pozici kurzoru.
SC = 1 - přesunovat znaky.
SC = 0 - přesunovat kurzor.
RL = 1 - přesun vpravo.
RL = 0 - přesun vlevo.
DL = 1 - osmibitový přenos.
DL = 0 - čtyřbitový přenos.
N = 1 - dvojitá linka.
N = 0 - jednoduchá linka.
Acg - adresa nastavení čítače CG RAM.
Add - adresa nastavení čítače DD RAM.
BF - flag BUSY.
BF = 1 - displej zaneprázdněn.
BF = 0 - displej připraven přijímat instrukce.
AC - adresový čítač.

Ovládání LED

Čtyři LED D9 až D12 se ovládají zápisem čtyřbitových dat D0 až D3 na adrese 05000H. Při počáteční inicializaci je nutné zapsat na této adrese data 0XFH, přičemž po této operaci LED zhasnou. Na místo X lze zapsat libovolná data. LED se rozsvítí zápisem 0 na příslušnou pozici (D0 až D3).

Ovládání akustické signalizace (BEEP)

Akustická signalizace se aktivuje zápisem na adresu 04800H. Přitom nezáleží, jaká data budou zapsána, protože signalizace proběhne pouze součinností stavu dekodérů adres se signálem WR\. Délka akustické signalizace je přednastavena hardwarově časovou konstantou R1, C10 a lze ji prodloužit opakovaným zápisem na tuto adresu v určitých časových prodlevách. Dobu časové prodlevy je nutno odzkoušet, protože délka vlastní akustické signalizace závisí na toleranci součástek (rezistoru R1 a kondenzátoru C10).

Adresová tabulka

Adresy pro ovládání periferií jsou shrnuty v tab. 5. Adresy jsou vztaženy k dekodéru na desce BASIC 552 v součinnosti se signálem CS3. V tabulce jsou uvedeny pouze základní adresy, stínové adresy vzhledem k úrovním jednotlivých adresových bitů je nutno respektovat. Aktivovat BEEP lze tudíž nejen na adrese 04800H, ale na všech adresách, u nichž je bit A11 ve stavu log. 1 (např. od 04800H až po 04FFFH apod.).

Volný adresový prostor

S ohledem na obvodové řešení není vhodné využívat signál CS3 z desky BASIC pro ovládání jiných desek, s výhodou lze využít signálů CS4 a CS5 nebo sériového styku I2CBUS. Pokud však chceme aktivovat jiné zařízení právě signálem CS3, nabízí se volný prostor od adresy 06008H do 067FFH. Na další adrese je již aktivována akustická signalizace BEEP.

Stavba

Na obr. 5 jsou obrazce plošných spojů na straně pájení a na straně součástek a na obr. 6 je rozmístění součástek na desce. Deska je s oboustrannými spoji a s prokovenými otvory.

Displej je přišroubován k desce pomocí distančních sloupků a vodiče jsou propojené konektorem. Nad spodní hranu desky displeje je položena klávesnice, která je s deskou

KEYDSP1 rovněž spojena konektorem. Indikační LED jsou povysunuty do takové vzdálenosti od desky, aby při montáži s jakýmkoliv čelním panelem vyčnívaly nad jeho povrch.

Všechny integrované obvody jsou kvůli spolehlivosti zapájeny bez objímek přímo do desky. Před pájením však musíme důkladně zkontrolovat správnou orientaci součástek, protože při jejich vyjímání v amatérských podmínkách se mohou spoje na desce snadno poškodit. Pokud už však kteroukoliv ze součástek osadíme nesprávně, je vhodné ji vyjmout pomocí horkovzdušné pistole s regulovanou teplotou výstupního vzduchu. Teplota musí být nastavena tak, aby se cín tavil a přitom výrazně nezměkl základní nosný materiál desky.

Nejprve osadíme konektory X2 a X3 tak, že je vložíme ze stany pájení a zapájíme ze strany součástek.

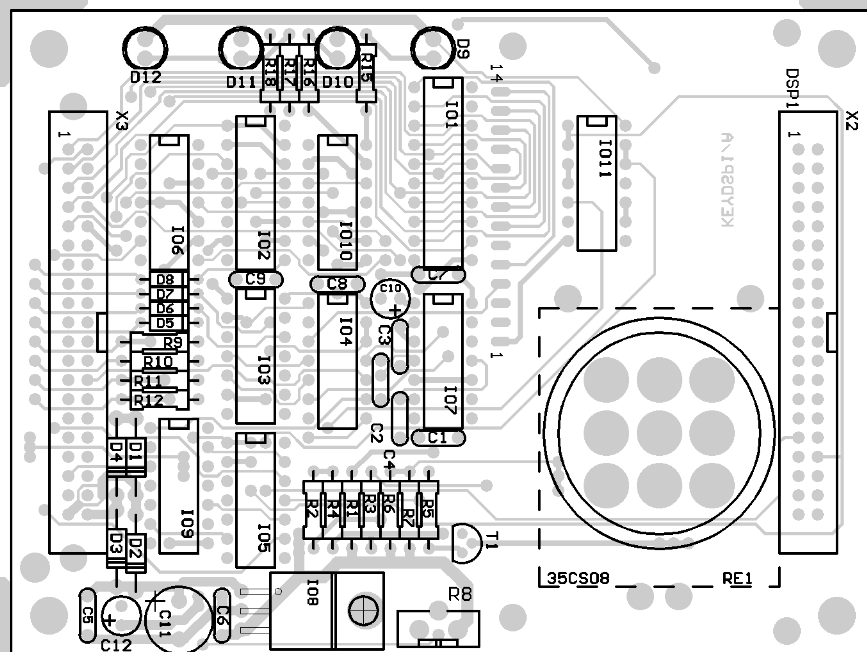
Potom osadíme a zapájíme R1 až R12 (kromě trimru R8), D1 až D8 a stabilizátor IO8, který je položen na desku a připevněn k ní šroubem M3x10 s obyčejnou a pérovou podložkou a maticí. Pokračujeme osazením C1 až C12, trimru R8, tranzistoru T1 a IO1 až IO11. C10 je položen na desku směrem k LED mezi obvody IO1 a IO10.

K desce s plošnými spoji modulu displeje je nutné před jeho instalací připájet jednořadý konektor, který vznikne rozlomením kontaktní lišty S1G20. Takto vzniklou část lišty se čtrnácti kolíky vložíme ze strany pájení do desky displeje a připájíme ji ze strany součástek. Do desky KEYDSP1 vložíme dutinkovou lištu BL815G tak, že nejprve z ní vyjmeme krajní dutinku a takto upravenou lištu zasadíme do desky ze strany součástek (odstraněnou dutinkou směrem k tranzistoru T1) a zapájíme. Mezi desku KEYDSP1 a desku displeje vložíme čtyři distanční sloupky délky 12 mm (KDR12) a celek sešroubujeme čtyřmi šrouby M3x20 s válcovou hlavou. Šrouby vložíme ze strany displeje a matice přišroubujeme ze strany pájení na desce KEYDSP1. Přitom musíme dbát na to, aby byl konektor modulu displeje správně zasunut do konektoru na desce.

Podobným způsobem je osazena i klávesnice. Z dutinkové lišty BL810G s deseti dutinkami se musí vyjmout obě krajní dutinky. Lištu vložíme do desky KEYDSP1 ze strany součástek tak, aby byla vystředěna vzhledem k průchodům na desce, a připájíme ji ze strany pájení. Konektorové kolíky ASS01038Z nejprve rozlomíme na část s osmi kolíky a odstraníme štípacími kleštěmi horní rozpěrku z plastické hmoty (tu, u které jsou kratší vývody). Do děr na bocích konektoru X3 přišroubujeme k desce dva distanční sloupky DI5M3x15. Sloupky jsou umístěny na straně součástek a jsou přišroubovány dvěma šrouby M3x10 ze strany spojů. Do lišty BL810G vložíme upravené konektorové kolíky ASS01038Z a z horní strany desky KEYDSP1 přiložíme klávesnici tak, aby dosedla na distanční sloupky. Konektorové kolíky do

Tab. 5. Adresová tabulka

Adresa	Činnost
04800H	BEEP - aktivace odskokem na tuto adresu
05000H	LED - zápis na D0 až D3
06000H	KLÁVESNICE - čtení 1.
06001H	KLÁVESNICE - čtení 2.
06002H	KLÁVESNICE - čtení 3.
06003H	KLÁVESNICE - čtení 4.
06004H	DISPLEJ - zápis řídicí instrukce
06005H	DISPLEJ - zápis dat
06006H	DISPLEJ - čtení flagu BUSY a adresy
06007H	DISPLEJ - čtení dat



Obr. 6. Rozmístění součástek na desce klávesnice a displeje KEYDSP1

desky klávesnice zapájíme. Distanční sloupky slouží pouze k vymezení vzdálenosti klávesnice od desky KEYDSP1, ke klávesnici nejsou přišroubovány.

Klávesnice se opírá o spodní okraj displeje a v závěrečné montáži je ji možné připevnit k desce displeje oboustrannou samolepicí fólií tloušťky 1 mm. Při montáži k čelnímu panelu bude poloha klávesnice definována otvorem, který musí kopírovat zapuštěnou horní hranu klávesnice.

LED D9 až D12 vložíme do desky tak, aby vzdálenost mezi deskou a tělem LED byla 16 mm. Pak vývody LED připájíme.

Reproduktor osadíme na desku ze strany pájení. Vývody reproduktoru zkrátíme na délku 3 mm od spodního okraje pouzdra, reproduktor přiložíme k desce a připájíme k ploškám.

Nakonec přišroubujeme čtyři distanční sloupky KDI6M3x20 do rohových děr na desce.

Oživení

Po osazení součástek nejprve zkontrolujeme, zda nejsou plošné spoje propojené cínovými můstky, které mohou vzniknout při pájení. Na napájecí vývody konektoru X3 přivedeme stejnosměrné napětí 9 až 12 V z externího zdroje (nezáleží na polaritě) a voltmetrem zkontrolujeme přítomnost napětí 5 V na příslušných vývodech jednotlivých integrovaných obvodů při vyjmuté klávesnici.

Trimr R8 vytočíme zcela vlevo a potom jím otáčíme pomalu doprava tak, aby na displeji byly zřejmé mírně tmavé plošky na místech jednotlivých znaků. Klávesnici vložíme zpět.

Před dalším oživováním je nutné spojit desku KEYDSP1 se základní deskou BASIC 552, protože jinak není možné simulovat zápis a čtení dat. Odpojíme napájecí napětí a desku zasuneme konektory X2 a X3 do konektorů X4 a X5 na desce BASIC 552. **POZOR!** Desku nesmíme otočit, je orientována tak, že mikroprocesor 80C552 na desce BASIC je pod LC displejem! Obě desky spolu sešroubujeme prostřednictvím rohových distančních sloupků.

Pokud budeme systém oživovat ve spolupráci s emulátorem, lze desky propojit plošnými vodiči, které budou na koncích opatřeny samořeznými dutinkovými konektory PFL40, do nichž jsou vloženy dvouřadé konektorové kolíky S2G40. K propojení vodiči je potom nutno využít konektorů X2 a X3 na desce BASIC 552. Před připojením napájecího napětí zkontrolujeme ohmmetrem souhlasnost signálů na jednotlivých vývodech konektorů na obou deskách.

V oživování postupujeme dále zápisem dat na příslušné adresy, čímž odzkoušíme činnost displeje, LED a akustické signalizace BEEP.

Po zápisu znaků na displej dostavíme trimr R8 tak, aby znaky byly zřetelné, a aby na prázdných místech nebyly viditelné tmavé stopy pozadí.

Seznam součástek

R1, R2, R9, R10, R11, R12	10 kΩ, miniaturní
R3	680 Ω, miniaturní
R4, R15, R16, R17, R18	2,2 kΩ, miniaturní
R5	1 kΩ, miniaturní
R6, R7	15 Ω, miniaturní
R8	5 kΩ, trimr PT10H
C1, C2, C5, C6, C7, C8, C9	100 nF, keramický
C3, C4	10 nF, keramický
C10, C12	10 µF/35 V, rad.
C11	470 µF/25 V, rad.
D1, D2, D3, D4	1N4007
D5, D6, D7, D8	BAT46
D9, D10, D11, D12	LED R, 5 mm, 2 mA
T1	BC639
IO1	74HCT245
IO2	74HCT155
IO3	74HCT20
IO4, IO5, IO11	74HCT00
IO6	74HCT126
IO7	NE556
IO8	7805
IO9	74HCT02
IO10	74LS77
DSP1	MC1602E
RE1	35CS08
X1	ASS01038Z
X2, X3	ASS24038Z
dutinková lišta BL810G	1 kus
dutinková lišta BL815G	1 kus
kontaktní lišta S1G20	1 kus
distanční sloupek KDI6M3x20	4 kusy
distanční sloupek DI5M3x15	2 kusy
distanční sloupek KDR12	4 kusy
deska s plošnými spoji KEYDSP1	

PORT 64 - deska rozšiřující počet portů

Popis zapojení

Deska PORT 64 rozšiřuje počet portů základní desky BASIC 552. Schéma je na obr. 7. Na desce je 8 obvodů typu PCF8574, které rozšiřují počet vstupně/výstupních portů mikropočítače o 64 linek a obvod PCF8591, který poskytuje čtyři osmibitové analogové vstupy a jeden osmibitový analogový výstup.

Komunikace mezi mikropočítačem 8xC552 na desce BASIC 552 a obvody na desce PORT 64 probíhá podle protokolu sběrnice I2C. Tato sběrnice představuje sériové dvouvodičové (dvoulinkové) spojení mezi obvody.

Funkce sběrnice I2C a její ovládání budou popsány v kapitole o mikropočítači 80C552.

Zapojení je poměrně jednoduché, linky SCL, SDA a INT jsou spojené paralelně. Linky SCL a SDA tvoří sběrnici I2CBUS. Po lince SCL jsou přenášené hodinové impulsy a po SDA jsou přenášena data. Výstupy INT od obvodů PCF8574 jsou s otevřenými kolektory, úroveň log. 1 zabezpečuje rezistor R1. Činnost těchto obvodů je stručně popsána dále.

Obvod PCF8591 má na analogových vstupech ochranné rezistory R3 až R6.

Pro správnou činnost převodníku je nutno zajistit přesné referenční napětí na vstupu Uref. O to se stará obvod IO10 typu LM336, což je referenční dioda 2,5V. Protože vstup Agnd je spojen s napájecí zemí, jsou k tomuto potenciálu vztaženy všechny analogové vstupy a analogový výstup.

Místo referenční diody IO12 lze osadit diodu s jiným napětím až do 4,5 V. I když je možné na vstup Uref připojit referenční napětí až 5 V, v žádném případě nesmí být tento vstup propojen s napájecím napětím Vcc. Napájecí větev nemusí být stabilní (mohou se na ní vyskytovat rušivé signály) a hodnota po ukončení převodu by byla značně zkreslena. Kondenzátor C5 filtruje napětí za referenční diodou.

Napájecí napětí je na desku PORT 64 přivedeno z desky BASIC 552 a je filtrováno kondenzátory C2 až C4.

Obvod PCF8574

Obvod rozšiřuje počet portů mikropočítače. Lze jej jednoduše připojit ke sběrnici I2CBUS linkami SDA a SCL.

Obvod je vyroben technologií CMOS. Potřebuje jediné napájecí napětí v rozsahu 2,5 až 6 V a v klidovém stavu má spotřebu 10 μA. Obvod obsahuje osm vstupně/výstupních portů (I/O porty), které mohou být využity jak pro zápis, tak pro čtení.

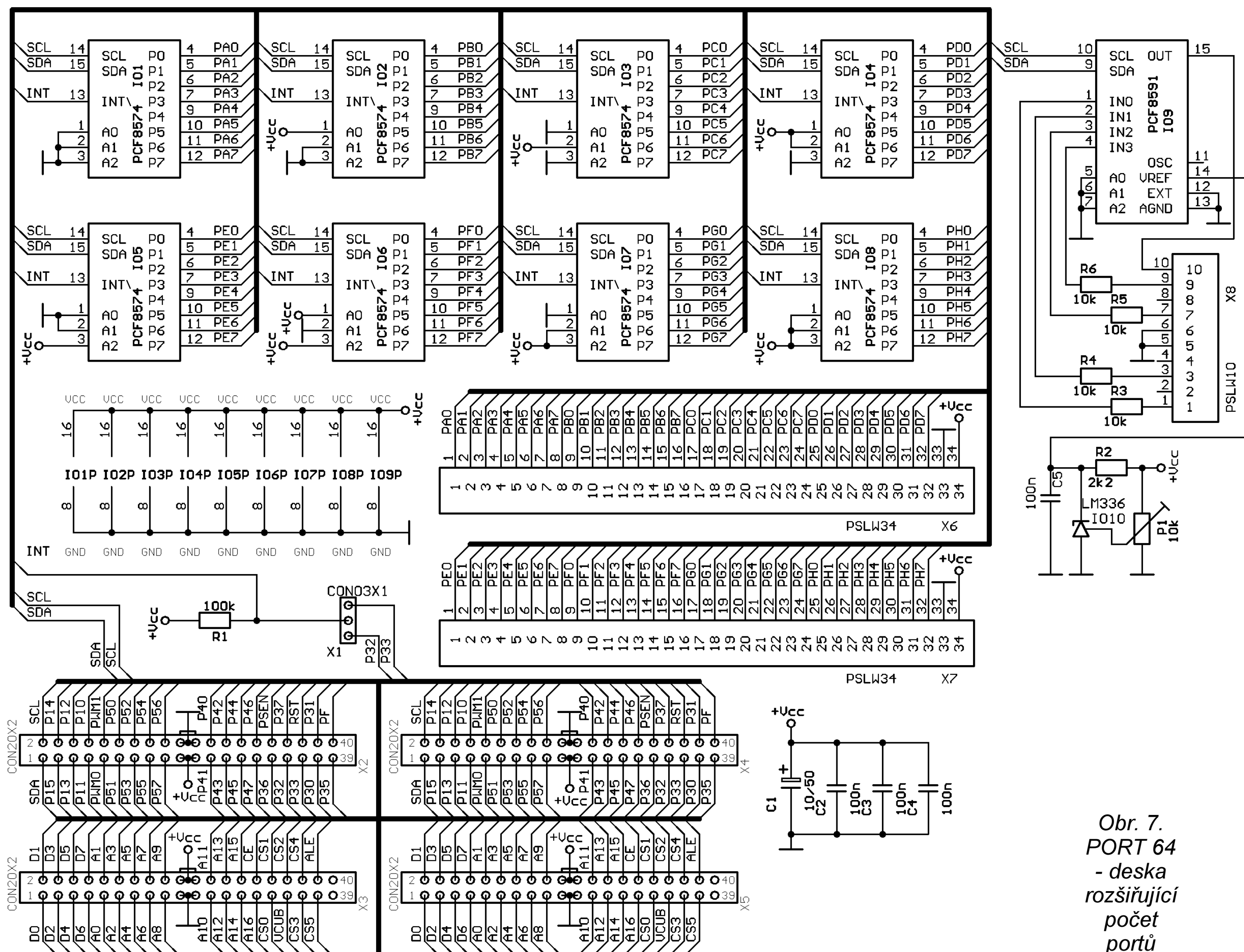
Dále obvod obsahuje tři adresové linky a výstup pro žádost o přerušení. Pevně přednastavená bázová adresa je 32, na třech adresových linkách lze vně obvodu adresovat až osm těchto stejných obvodů. Zapojení vývodů pouzdra IO je v tab. 5.

Výstup INT\ žádosti o přerušení je s otevřeným kolektorem, takže lze výstupy několika obvodů zapojit paralelně. Mezi výstupy a napájecí sběrnici je zapojen zdvihací (pull-up) rezistor, který na výstupech udržuje klidovou vysokou úroveň H. Pokud je obvod ve vstupním modu, vygeneruje žádost o přerušení při sestupné nebo vzestupné hraně signálu na kterémkoliv I/O portu. Žádost o přerušení se projeví uvedením výstupu INT\ do nízké úrovně L. V průběhu čtení linka opět přejde do úrovně H.

I/O porty mají po zapnutí napájecího napětí a v klidovém stavu úroveň H a velký odpor, takže mohou být uvedeny vnějšími obvody do úrovně L. Každý z IO portů může být ovládán interním nízkoohmovým kolektorovým výstupem, takže může být interně uveden do úrovně L. Celá osmice portů může být tedy částečně využita pro zápis a částečně pro čtení. Pouze čtecí porty musí být nastaveny do úrovně H, protože jinak by nebylo možné jejich stav číst.

Tab. 5. Zapojení vývodů obvodu PCF8574

Vývod	Symbol	Popis
1	A0	adresový hardwarový vstup A0
2	A1	adresový hardwarový vstup A1
3	A2	adresový hardwarový vstup A2
4	P0	I/O port 0
5	P1	I/O port 1
6	P2	I/O port 2
7	P3	I/O port 3
8	Uss	napájecí zem
9	P4	I/O port 4
10	P5	I/O port 5
11	P6	I/O port 6
12	P7	I/O port 7
13	INT\	výstup přerušení (aktivní nula)
14	SCL	sériové hodiny
15	SDA	sériová data
16	Udd	napájecí napětí



Obr. 7. PORT 64 - deska rozšiřující počet portů

Tab. 6. Zapojení vývodů obvodu PCF8591

Vývod	Symbol	Popis
1	IN0	analogový vstup 0
2	IN1	analogový vstup 1
3	IN2	analogový vstup 2
4	IN3	analogový vstup 3
5	A0	adresový hardwarový vstup A0
6	A1	adresový hardwarový vstup A1
7	A2	adresový hardwarový vstup A2
8	Uss	napájecí zem
9	SDA	sériová data
10	SCL	sériové hodiny
11	OSC	oscilátor vstup/výstup
12	EXT	externí/interní spínání pro vstup oscilátoru
13	AGND	analogová zem
14	UREF	vstup referenčního napětí
15	OUT	analogový výstup (D/A převodník)
16	Udd	napájecí napětí

Obvod PCF8574 spolupracuje s rozhraním I2CBUS jako obvod SLAVE. Obvod se může nacházet ve dvou stavech, z nichž první je mod zápisu (zápis dat na I/O linky) a druhý je mod čtení (čtení stavů I/O linek).

Obvod PCF8591

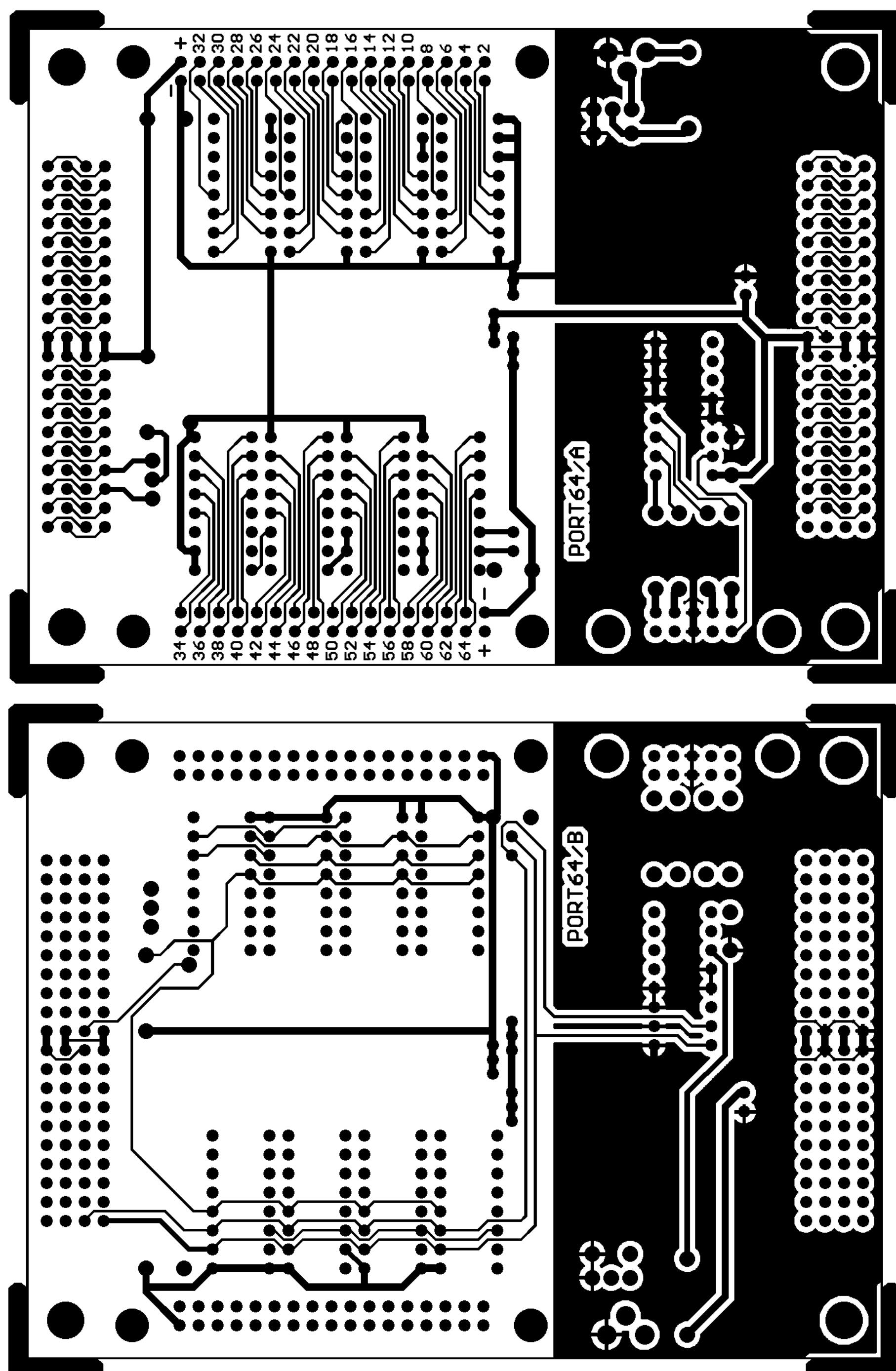
Obvod obsahuje čtyři analogové vstupní kanály a jeden analogový výstupní kanál. Lze jej jednoduše připojit ke sběrnici I2CBUS linkami SDA a SCL. Je vyroben technologií CMOS, má jediné napájecí napětí v rozsahu 2,5 až 6 V a spotřebu v klidovém stavu 15 µA.

Rozlišovací schopnost všech kanálů je osm bitů. Pevně přednastavená bázová adresa je 72, na třech adresových linkách lze vně obvodu adresovat až osm těchto stejných obvodů. Zapojení vývodů pouzdra obvodu je v tab. 6.

Řídicí byte. Obvod PCF8574 je nutno předem nakonfigurovat do požadovaného režimu. Řídicí byte je druhý byte, který se vysílá po adresování obvodu. Tento byte je zapsán do řídicího registru. Horní čtyři bity (od MSB) slouží k uvolnění analogového výstupu a ke konfiguraci čtyř vstupních analogových linek jako samostatných nebo diferenciálních vstupů. Dolní čtyři bity slouží k výběru analogových vstupních kanálů a k nastavení příznaku automatického inkrementování. Pokud je příznak automatického inkrementování nastaven, je číslo výběru kanálu automaticky zvýšeno o jednotku po každém A/D převodu. Pokud je naadresován nejvyšší kanál a nastane inkrementace, následující vybraný kanál bude 0. Nejvyšší bity každé poloviny byte (high nibble a low nibble) jsou určeny pro budoucí funkce a musí být nastaveny vždy do stavu log. 0.

Tab. 7. Řídicí byte obvodu PCF8574

Bit	Funkce
7	vždy 0
6	uvolnění analogového výstupu (aktivní ve stavu log. 1)
5 a 4	zapojení analogových vstupů 00 - čtyři samostatné analogové vstupy 01 - tři diferenční vstupy AIN0 až AIN2 proti AIN3 (AIN3 invertující) 10 - dva samostatné analogové vstupy AIN0 a AIN1 a jeden diferenční vstup AIN2 proti AIN3 (AIN2 neinvert.) 11 - dva diferenční vstupy AIN0 proti AIN1 a AIN2 proti AIN3 (AIN0 a AIN2 neinvertující)
3	vždy 0
2	bit aut. inkrementace (aktivní v log. 1)
1 a 0	číslo A/D kanálu 00 - kanál 0, 01 - kanál 1, 10 - kanál 2, 11 - kanál 3



Obr. 8. Obrazec plošných spojů na straně A - straně pájení (nahoře) a na straně B - straně součástek (dole) desky rozšiřující počet portů PORT 64 (měř.: 1 : 1)

Po připojení napájecího napětí nebo po vynulování jsou všechny bity řídicího registru ve stavu log. 0. Analogový výstup je potom připojen na vysokou úroveň s vysokou impedancí.

Význam jednotlivých bitů řídicího byte je v tab. 7.

D/A převod je aproximační. Interní DA převodník je připojen svým odporovým děličem na zdroj referenčního napětí. Výstupní napětí převodníku může nabývat 256 úrovní. Po převodu je připojeno napětí z odporového děliče na výstup AOUT.

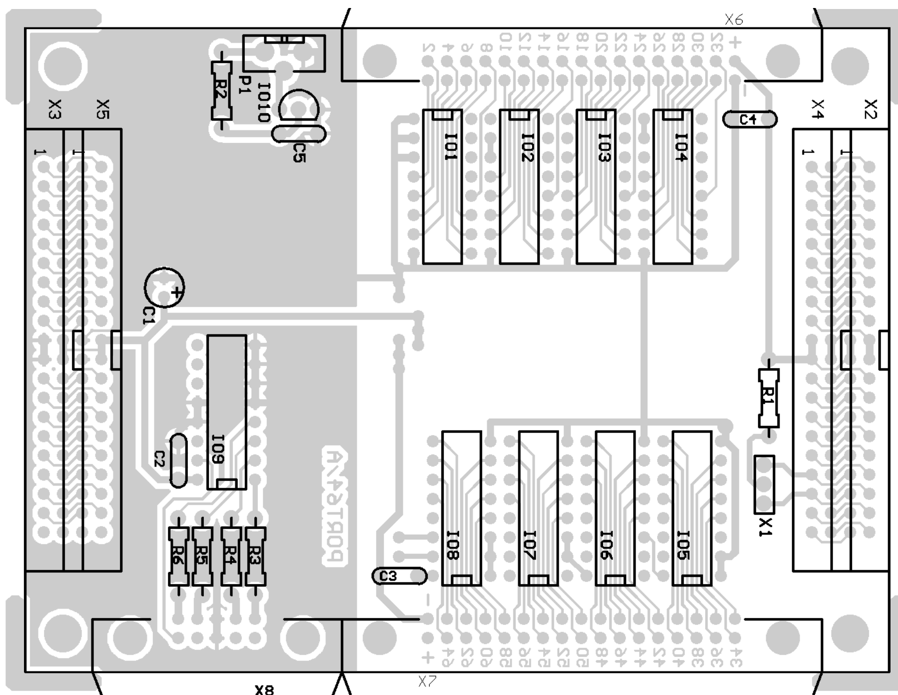
Na výstupu převodníku je oddělovací zesilovač, který může být zapnut nebo vypnut. Stav zesilovače se ovládá nastavením bitu číslo 6 (uvolnění analogového výstupu) v řídicím registru. Oddělovací zesilovač také zachovává svoji poslední výstupní úroveň v době dalšího převodu.

Pro A/D převod je využíván stejný aproximační registr a převodník, jako pro D/A převod. A/D převodní cyklus je vždy nastartován po přijetí platného čtecího modu a je spouštěn závěrnou hranou potvrzovacího (acknowledge) bitu.

Vzorek vstupního napětí je odebrán z vybraného analogového vstupu (nebo z diferenčního vstupu podle nastavení řídicího registru), zapamatován a převeden na osmibitový binární kód.

Po převodu je hodnota uložena v interním registru ADC. Další kanál se vybere po převodu automaticky, pokud je nastaven bit automatické inkrementace v řídicím registru.

A/D převod je taktován vnitřním hodinovým oscilátorem. Při použití vnitřního oscilátoru musí být vývod EXT spojen se sběrnicí Uss. Pokud je sběrnice Uss spojena s napájecí sběrnicí Udd, je výstup oscilátoru ve stavu vysoké impedance a k vývodu OSC je možné připojit vnější taktovací signál.



Obr. 9. Rozmístění součástek na desce rozšiřující počet portů PORT 64

Stavba

Na obr. 8 jsou obrazce plošných spojů na straně pájení a na straně součástek, na obr. 9 je rozmístění součástek na desce. Deska je s oboustrannými spoji s prokovenými otvory.

Nejprve osadíme všechny rezistory R1 až R6, kondenzátory C1 až C5, zkratovací kolíky X1, objímky pro integrované obvody IO1 až IO19, referenční diodu IO10 a konektory X6 až X8. Dále osadíme trimr P1. Na pozice konektorů X2 až X5 osadíme příslušné konektory (lištové a dutinkové) tak, aby bylo možné desku spojit s deskou BASIC 552, popř. s další deskou, která může být umístěna pod desku PORT 64. Po propojení s deskou BASIC 552 zkontrolujeme ohmmetrem souhlasnost stejných signálů na obou kartách.

Oživení

Desku oživíme ve spolupráci s deskou BASIC 552, ze které je deska PORT 64 napájena a řízena. Po zapnutí zkontrolujeme napájecí napětí na všech objímkách integrovaných obvodů a trimrem P1 nastavíme referenční napětí na požadovanou velikost. S použitou referenční diodou lze napětí regulovat v rozsahu asi 2,4 až 3,9 V.

Tím je ukončeno základní oživení, další kroky mohou být provedeny po napsání příslušného programu, který umožní komunikaci s jednotlivými obvody IO1 až IO9 po sběrnici I2CBUS. Vsunutím propojky na zkratovací kolíky X1 je možné přiřadit hardwarové žádosti o přerušení od IO1 až IO8 vstupům INT0 nebo INT1 mikropočítače 8xC552. Je nutné si uvědomit, že tyto žádosti o přerušení nesmí kolidovat s žádostmi o přerušení od obvodů karty BASIC 552.

Upozornění! Při spojování desek PORT 64 a BASIC 552 je nutné dávat pozor na orientaci obou desek. Na vstupech PCF8574 nejsou ochranné obvody, vstupní napětí na každý jednotlivý port nesmí překročit hodnotu 5 V!

Seznam součástek

R3	680 Ω, miniaturní	
R1	100 kΩ, miniaturní	
R2	2,2 kΩ, miniaturní	
R3 až R6	10 kΩ, miniaturní	
P1	10 kΩ, trimr PT10H	
C1	10 µF/50 V, rad.	
C2 až C5	100 nF, keram.	
IO1 až IO8	PCF8574	
IO9	PCF8591	
IO10	LM336	
X1	S1G20	
X2, X3	ASS24038Z	
X4, X5	BL840GD	
X6, X7	PSL34W	
X8	PSL10W	
objímka DIL16PZ		8 kusů
zkratovací propojka JUMP-RT		1 kus
deska s plošnými spoji č.: PORT64		

Závěr konstrukční části

Desky s plošnými spoji popisovaných konstrukcí jsou s oboustrannými spoji s prokovenými otvory, takže nepřipadá v úvahu jejich amatérská výroba. Samostatné desky nebo celé zkompletované stavebnice se součástkami si však můžete objednat u autora na telefonním čísle 0606 358403.

Programy je nuto napsat buď textově a přeložit vhodným assemblerem nebo je možné využít vyšších programovacích jazyků, např. jazyk „C". Vývoj programu může probíhat za podpory simulátoru EPROM, který je možné programovat běžnými programátory nebo může být použit emulátor s příslušným softwarem. Emulátory si můžete též objednat na uvedeném telefonním čísle, nejsou však levnou záležitostí. Jejich cena se pohybuje mezi 10 000 až 25 000 Kč, záleží na typu.

Veškeré ceny budou sděleny na telefonický dotaz.

Popis mikropočítače 80C51

Aby bylo možné pochopit činnost náročného mikropočítače 80C552, je nutné nejdříve vysvětlit funkci mikropočítače 80C51.

Mikropočítač 80C51 je osmibitový jednočipový mikrořadič. Jeho vnitřní zapojení a funkce je popsána v následujících kapitolách.

Mikropočítače řady „51" mají oddělenou paměť programu a paměť dat. Z žádné paměti dat nelze spustit program podobně, jako např. u Z80. Kapacita obou pamětí může být max. 64 kbyte.

Zapojení vývodů obvodů 80C51 a 80C52 je na obr. 10. Vnitřní uspořádání mikropočítačů řady 80C51 je na obr. 11.

Datové brány

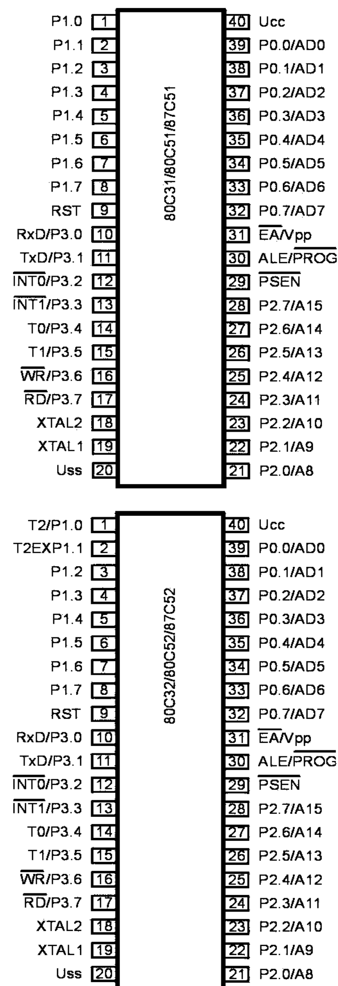
Mikropočítač obsahuje čtyři obousměrné datové brány (porty). Port je možno chápat jako celek (např. P1 je osmibitový port) nebo jako jednotlivé linky celkového portu (např. port P1.0 je první linka portu P1 a port P1.7 je poslední osmá linka portu P1). Porty se taktéž nazývají datové brány. Zapojení struktur datových bran 0 až 3 je na obr. 12 až obr. 15.

Protože datové brány jsou obousměrné, lze do každého portu (např. linky P1.0) informaci zapisovat nebo z ní číst. Každá brána obsahuje výstupní vyrovnávací paměť, která je adresovaná jako speciální funkční registr SFR, vstupní hradlo a výstupní budič.

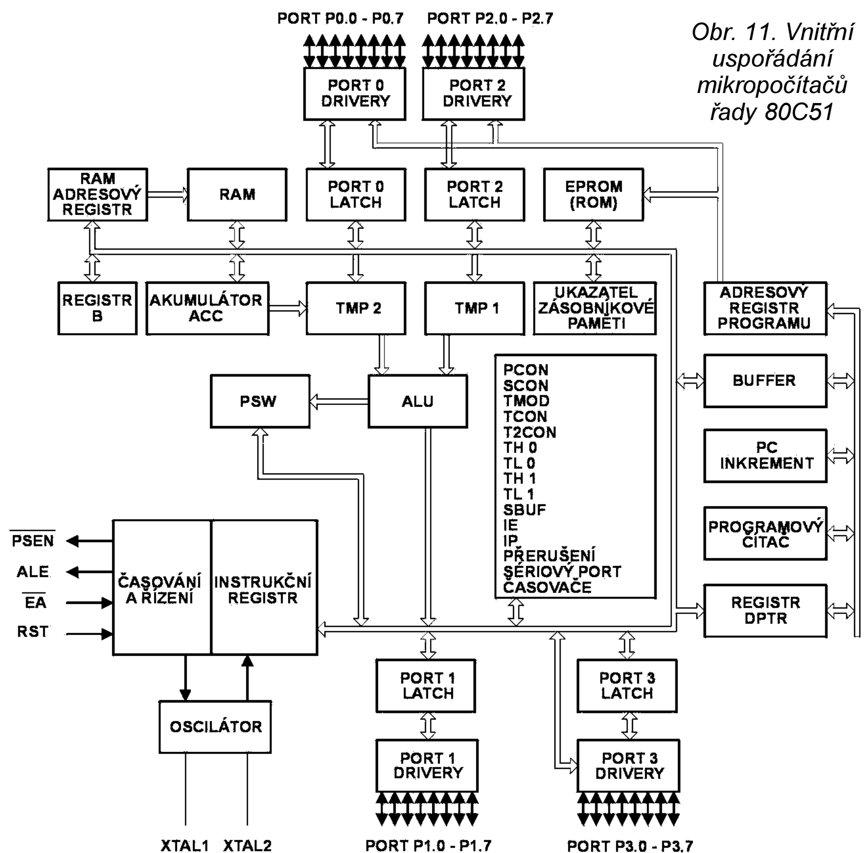
Výstupní budiče bran P0 a P2 mohou být připojeny na vnitřní adresovou a datovou sběrnici mikropočítače řídicím signálem CTRL. Proto se tyto dvě brány používají k připojení vnějších pamětí programu a dat. Na bráně P0 se vytvoří vnější multiplexovaná adresová a datová sběrnice a brána P2 generuje horní část adresové sběrnice. Během přístupu k vnější paměti se nemění obsah výstupní vyrovnávací paměti příslušné brány. Výstupní budič brány P0 má v režimu sběrnice definované stavy nízké a vysoké úrovně. Pokud však pracuje jako budič datové brány, stává se výstupním budičem s otevřenými kolektory, a proto je nutné doplnit k jednotlivým vývodům rezistory, připojené k napájecí sběrnici mikropočítače. Výstupní budič brány P2 tyto rezistory obsahuje ve své struktuře.

Výstupní budič brány P3 má sice předepnuté výstupy rezistory na napájecí napětí (rezistory pull-up), avšak signály, které vstupují do tohoto budiče, jsou logickým součinem obsahu vyrovnávací paměti datové brány P3 a řídicích bitů mikropočítače. Proto, pokud se některé z řídicích bitů využívají, musí se zajistit na odpovídajícím místě registru P3 hodnota log. 1. Jinak v logickém součinu bude příslušný řídicí bit nastaven do stavu log. 0 bitem datové vyrovnávací paměti bez ohledu na stav řídicího signálu.

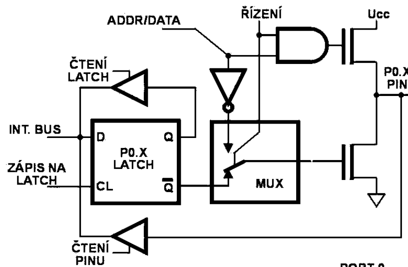
Při každém čtení obsahu datové brány se vyhodnocuje nejen stav na vývodu této brány, ale také obsah výstupní vyrovnávací paměti. Proto bit, který má být použitý jako vstupní ke čtení, musí mít na odpovídajícím místě výstupní vyrovnávací paměti programově zapsanou jedničku. Při čtení tohoto bitu se provádí logický součin vstupního signálu a obsahu příslušného bitu vyrovnávací paměti a výsledek je vyhodnocen jako čtení bitu registru Px. Při zápisu dat 00H do brány P3 se nastaví její vývody na nulovou úroveň a žádný řídicí signál tuto úroveň nemůže změnit. Výjimku při čtení datových bran tvoří instrukce, které čtou obsah brány, mění přečtené slovo a zpět jej vrací na původní místo. Tyto instrukce vyvolávají pouze čtení ob-



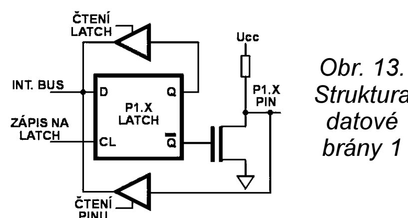
Obr. 10. Zapojení vývodů mikropočítače 80C51 (nahoře) a 80C52 (dole)



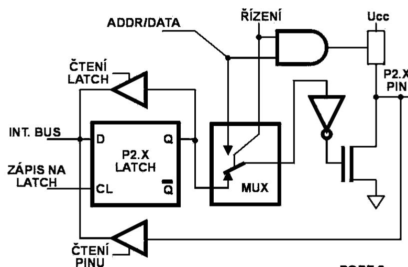
Obr. 11. Vnitřní uspořádání mikropočítačů řady 80C51



Obr. 12. Struktura datové brány 0



Obr. 13. Struktura datové brány 1



Obr. 14. Struktura datové brány 2

sahu výstupní vyrovnávací paměti bez ovlivnění skutečným stavem na vývodech brány. Jsou to např. instrukce ANL, ORL, XRL, JBC, CPL apod.

Každá linka brány P1, P2 a P3 může řídit 4 vstupy LS, linky brány P0 s doplněnými zdvihacími (pull-up) rezistory mohou řídit až 8 vstupů LS. Brány mohou být řízeny i obvody s otevřeným kolektorem, protože předepnutí vstupů na napájecí napětí plně postačí k definici log.1.

Některé linky (porty) bran vykonávají kromě základní funkce ještě funkci alternativní. Přehled alternativních funkcí je v tab. 8.

Paměť programu

Mikropočítač vykonává instrukce, které jsou ve formátu slov uloženy v paměti programu. Paměť programu podle typu IO může být integrována buď přímo na čipu (u typů 89C5x paměť FLASH a u typů 87C5x paměť EPROM) nebo je tvořena externí pamětí (většinou EPROM). Přístup k vnější paměti programu je zásadně prováděn signálem PSEN (non) a adresou, která je generována na portu P2 a na portu P0 multiplexovaně s datovou sběrnicí. To znamená, že nejprve je na portu P0 vystavena spodní část adresy (osm bitů) a signálem ALE (addres latch enable) je tato část zapsána do vnějšího adresového záchytného registru (LATCH). Potom je celá šestnáctibitová adresa přítomna částí v adresovém záchytném registru (A0 až A7) a částí na portu P2 mikropočítače (A8 až A15). Následně jsou na portu P0 zpracovávána data. Jako adresový záchytný registr se používají větši-

Obr. 15. Struktura datové brány 3

nou obvody 74HCT573, které jsou pro tuto činnost přizpůsobeny (zápisový vstup C se spojí se signálem ALE mikropočítače a vstup OE (output enable) se spojí s napájecí zemí.

Pro paměť programu jsou možné dvě konfigurace. Buď je možné využít pouze vnitřní paměť programu (u typů 87C5x) a případně ji doplnit vnější pamětí až do kapacity 64 kbyte, nebo lze využít jen paměť vnější. Pokud vystačí u typů 87C5x vnitřní paměť programu, nemusí být vnější pamětí doplněna.

Paměť programu má vyhrazeno několik míst, na které odskakuje program při obsluze přerušení apod. Těmto paměťovým místům se říká rezervované adresy a jsou shrnuté v tab. 9.

Paměť dat

Každý mikropočítač řady „51" obsahuje vnitřní paměť dat RAM. Kromě této paměti má ještě skupinu speciálních funkčních registrů SFR pro řízení vnitřní struktury mikropočítače. Mimo jiné může být ještě adresováno 64 kbyte vnější paměti dat, kterou lze adresovat nepřímo pomocí vnitřních registrů R0 nebo R1, popř. přímo šestnáctibitovým registrovým párem DPTR, složeným ze dvou osmibitových registrů DPL a DPH.

Registry R0 až R7 mohou být v programu využity obecně ke zpracování dat. Funkci je nejlépe možné pochopit při studiu instrukčního souboru. Mikropočítač obsahuje čtyři banky registrů R0 až R7, takže je možné pracovat celkem se třiceti dvěma registry R. Banky se přepínají programově a v reálném čase lze pracovat vždy pouze s jednou bankou registrů.

Rozdělení vnitřní paměti dat ukazuje tab. 10. Z rozdělení adresového prostoru je vidět, že od adresy 00H do 7FH je mapa bitově adresovatelných míst a od adresy 80H jsou v paměti umístěny speciální funkční registry (SFR), které zahrnují nejenom řídicí a stavová slova, ale i datové brány pro paralelní a sériový výstup, výstup dat, akumulátor a registr B.

U mikropočítačů 8032 (52) je horní polovina vnitřní paměti dat přístupná pouze nepřímým adresováním.

Význam jednotlivých SFR bude popsán dále.

Tab. 8. Přehled alternativních funkcí portů mikropočítače 80C51

Port	Alt. funkce	Popis
P0.0 až P0.7	AD0 až AD7	Multiplexovaná adresová a datová sběrnice
P1.0	T2	Vstup pro časovač T2 (pouze u řady 80C52)
P1.1	T2EX	Obnova počítacího registru T2 (pouze u řady 80C52)
P1.4 až P1.7	-	Běžné I/O linky
P2.0 až P2.7	A8 až A15	Vyšší byte adresy
P3.0	RxD	Sériový vstupní port UART
P3.1	TxD	Sériový výstupní port UART
P3.2	INT0\	Externí žádost o přerušení 0
P3.3	INT1\	Externí žádost o přerušení 1
P3.4	T0	Externí vstup časovače T0
P3.5	T1	Externí vstup časovače T1
P3.6	WR\	Strobovací výstup pro zápis do externí paměti dat
P3.7	RD\	Strobovací výstup pro čtení z externí paměti dat

Tab. 9. Rezervované adresy v paměti programu mikropočítače 80C51

Adresa	Činnost
0000H	start systému, inicializace po vynulování (RESET)
0003H	odskok na tuto adresu při externím přerušení 0
000BH	odskok na tuto adresu při přeplnění časovače 0
0013H	odskok na tuto adresu při externím přerušení 1
000BH	odskok na tuto adresu při přeplnění časovače 1
0023H	obsluhuje sériový kanál
002BH	odskok na tuto adresu při přeplnění časovače 2

Přerušovací systém

Mikropočítače „51” využívají celkem pět zdrojů žádosti o přerušení programu (šest u 8032). Přerušení může být vyvoláno vnější žádostí na portech P3.2 nebo P3.3, a to buď úrovní, nebo sestupnou hranou vnějšího signálu podle nastavených bitů IT0 a IT1 řídicího registru TCON. Pokud je přerušení vyvoláváno úrovní, musí být aktivní úroveň přítomná po celou dobu obsluhy žádosti o přerušení.

Časovače generují samostatné příznaky TF0 a TF1 žádosti o přerušení.

Sériový kanál se hlásí společnou žádostí o přerušení od přijímače a vysílače.

Příznaky TF0 a TF1 se nulují automaticky při obsloužení. Všechny ostatní žádosti je nutné nulovat programově.

Každá žádost o přerušení má pevně stanovenou prioritu, kterou nelze měnit, ale lze vybrat z tohoto pořadí důležitější vstupy a přiřadit jim celkově vyšší prioritu registrem IP (interrupt priority). Registr IP je bitově adresovatelný.

Při generování požadavku na jakékoliv přerušení provede mikropočítač automaticky skok na rezervovanou adresu v paměti programu (viz tab. 9). Na této pozici musí být programem uskutečněn odskok do podprogramu obsluhy žádosti o přerušení. Návrat z obslužného podprogramu je nutno uskutečnit příkazem RETI, který obnoví původní obsah programového čítače a navíc oznámí, že končí obsluha žádosti o přerušení. Podobnou instrukcí RET se také obnoví původní obsah programového čítače, avšak není oznámen konec obsluhy žádosti o přerušení.

Na začátku podprogramu pro obsluhu přerušení se musí instrukcemi PUSH uschovat obsahy důležitých registrů do zásobníku, na konci obslužného podprogramu se pak instrukcemi POP přenesou data ze zásobníku zpět do registrů. Protože se jedná o postupné zapisování (zasouvání) dat do zásobníku, poslední data uložená instrukcí zápisu PUSH musí být načítána jako první instrukcí POP.

Příklad:

```
PUSH  Acc   ; uložení obsahu akumulátoru
PUSH  PSW   ; uložení obsahu stavového
              slova PSW
PUSH  B     ; uložení obsahu registru B
...obecné instrukce obslužného podprogramu...
POP   B     ; obnova obsahu registru B
POP   PSW   ; obnova obsahu stavového
              slova PSW
POP   Acc   ; obnova obsahu akumulátoru
```

Před odskokem do podprogramu pro obsluhu přerušení je vhodné vybrat taktéž jinou banku registrů R, budou-li v podprogramu tyto registry používány.

Pořadí priority signálů žádostí o přerušení je uvedené v tab. 11.

Bity registru IP mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
-	-	PT2	PS	PT1	PX1	PT	PX0

kde:

PT2 je priorita čítače 2,
PS je priorita sériového kanálu,
PT1 je priorita čítače 1,
PX1 je priorita vnějšího přerušení INT1 (na portu P3.3)
PT0 je priorita čítače 0
PX0 je priorita vnějšího přerušení INT0 (na portu P3.2)

Další registr, kterým se řídí přerušení, je registr IE (interrupt enable). Obsah tohoto registru povoluje nebo zakazuje jednotlivé žádosti o přerušení. Nastavením příslušného bitu do stavu log. 1 se žádost povoluje.

Bity registru IE mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

kde:

EA je celkové povolení přerušení
ET2 je povolení přerušení od čítače 2
ES je povolení přerušení od sériového kanálu
ET1 je povolení přerušení od čítače 1
EX1 je povolení přerušení od vnějšího přerušení INT1
ET0 je povolení přerušení od čítače 0
EX0 je povolení přerušení od vnějšího přerušení INT0

Při vyvolání přerušení sestupnou hranou vnějšího přerušovacího signálu musí být tento signál přítomen nejméně po dobu jednoho strojního cyklu, při vyvolání přerušení úrovní musí být přerušovací signál přítomen po celou dobu obsluhy přerušení.

Tab. 10. Rozdělení vnitřní paměti dat mikropočítače 80C51

Adresa	Popis	Zkratka
00H až 07H	banka registrů 0 (registry R0 až R7)	
08H až 0FH	banka registrů 1 (registry R0 až R7)	
10H až 17H	banka registrů 2 (registry R0 až R7)	
18H až 1FH	banka registrů 3 (registry R0 až R7)	
20H až 2FH	bitově adresovaný prostor	
30H až 7FH	volný prostor	
80H	port P0	P0
81H	ukazatel zásobníkové paměti	SP
82H	registr DPTR - dolní byte	DPL
83H	registr DPTR – horní byte	DPH
87H	registr PCON	PCON
88H	registr TCON	TCON
89H	registr TMOD	TMOD
8AH	počáteční hodnota čítače T0 - dolní byte	TL0
8BH	počáteční hodnota čítače T1 - dolní byte	TL1
8CH	počáteční hodnota čítače T0 - horní byte	TH0
8DH	počáteční hodnota čítače T1 - horní byte	TH1
90H	port P1	P1
98H	registr SCON	SCON
99H	registr sériového kanálu	SBUF
A0H	port P2	P2
A8H	registr IE	IE
B0H	port P3	P3
B8H	registr IP	IP
C8H	registr T2CON	T2CON
CAH	registr čítače 2 – dolní byte	CRAP2L
CBH	registr čítače 2 – horní byte	RCAP2H
CCH	počáteční hodnota čítače T2 - dolní byte	TL2
CDH	počáteční hodnota čítače T2 - horní byte	TH2
D0H	registr stavového slova	PSW
E0H	akumulátor Acc	A
F0H	registr B	B

Tab. 11. Pořadí priority signálů přerušení mikropočítače 80C51

Priorita	Zdroj
1	IE0
2	TF0
3	IE1
4	TF1
5	TI + RI
6	TF2 + EXF2

Čítače, časovače

Mikropočítač 8031 má dva a mikropočítač 8032 má tři šestnáctibitové čítače/časovače.

Pokud čítače/časovače pracují jako čítače, čítají vnější impulsy na portech T mikropočítače. Kmitočet vnějších impulsů musí být menší než kmitočet strojových cyklů.

Pokud čítače/časovače pracují jako časovače, zvyšuje se obsah registru v každém strojovém cyklu (čítá každých 12 period kmitočtu oscilátoru). Při použití krystalu 12 MHz má signál vstupující do čítače kmitočet 1 MHz.

Režim provozu čítačů/časovačů určuje registr TMOD.

Bity registru TMOD mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
1G	1CT	1M1	1M0	0G	0CT	0M1	0M0

kde bity D0 až D3 slouží pro nastavení čítače 0, bity D4 až D7 slouží pro nastavení čítače 1.

1G - ve stavu log. 1 je povolena činnost jen pro INT1 = TR1 = 1,
- ve stavu log. 0 je povolena činnost, pokud je TR1 = 1.

1CT - ve stavu log. 1 nastavuje vstup čítače na vývodu T1,
- ve stavu log. 0 nastavuje vstup na 1/12 kmitočtu oscilátoru.

0G - ve stavu log. 1 je povolena činnost jen pro INT0 = TR0 = 1,
- ve stavu log. 0 je povolena činnost, pokud je TR0 = 1.

0CT - ve stavu log. 1 nastavuje vstup čítače na vývodu T0,
- ve stavu log. 0 nastavuje vstup na 1/12 kmitočtu oscilátoru.

Význam bitů xM1 a xM0 je uveden v tab. 12. Registry TLx a THx (v praxi TL0 a TH0 nebo TL1 a TH1), uvedené v tab. 12, jsou počítací registry v oblasti RAM SFR. Režimy 0 až 3, uvedené v tab. 12, mají následující význam:

Režim 0. V tomto režimu jsou registry TLx a THx spřaženy sériově, přičemž TLx je pouze pětibitový. Celek potom se potom jeví jako třináctibitový čítač/časovač. Struktura časovače v tomto režimu je na obr. 16.

Režim 1. V tomto režimu jsou registry TLx a THx spřaženy sériově, přičemž TLx je plně osmibitový. Celek potom se potom jeví jako šestnáctibitový čítač/časovač.

Režim 2. Tento režim se využívá pro generování periodické žádosti o přerušení. TLx slouží jako osmibitový čítač/časovač, THx slouží jako paměť předvolby. Při každém přetečení počítacího registru se tudíž automaticky obnoví původní hodnota. Struktura časovače v tomto režimu je na obr. 17.

Režim 3. Zde pracuje pouze čítač/časovač 0, který je rozdělen na dva samostatné osmibitové bloky TL0 a TH0. TL0 používá klasické řídicí signály nultého čítače, včetně příznaku přetečení, TH0 využívá ze stojícího časovače 1 dva řídicí signály. Je to řídicí bit TR1 pro spuštění a zastavení časování a dále signál přetečení TF1. Struktura časovače v tomto režimu je na obr. 18.

Tab. 12. Význam bitů xM1 a xM0

xM1	xM0	Režim	Činnost
0	0	0	TLx slouží jako pětibitový dělič, THx pracuje jako osmibitový počítací registr
0	1	1	THx a TLx jsou sériově spřaženy a tvoří šestnáctibitový počítací registr
1	0	2	TLx slouží jako osmibitový čítač/časovač, THx slouží pro obnovu TLx při dočítání
1	1	3	TL0 je počítací registr osmibitového čítače/časovače se standardními řídicími signály čítače/časovače 0, TH0 je počítací registr osmibitového čítače/časovače se standardními řídicími signály čítače/časovače 1. Čítač/časovač 1 nečítá a čeká na jiný režim (než režim 3) činnosti

Obr. 16. Struktura čítače/časovače v režimu 0

Obr. 17. Struktura čítače/časovače v režimu 2

Obr. 18. Struktura čítače/časovače v režimu 3

Oba čítače/časovače jsou řízeny registrem TCON, jehož bity mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

kde:

TF1 je příznak přetečení čítače/časovače 1. Nastavuje se automaticky přetečením. Nuluje se automaticky při přechodu na adresu ošetření žádosti o přerušení od čítače/časovače 1. Může být nulován též programově, avšak potom se ruší žádost o přerušení od čítače/časovače 1.

TR1 je start/stop čítače/časovače 1. Ve stavu log. 1 čítač 1 čítá. Bit se nastavuje a nuluje pouze programově.

TF0 je příznak přetečení čítače/časovače 0. Nastavuje se automaticky přetečením. Nuluje se automaticky při přechodu na adresu ošetření žádosti o přerušení od čítače/časovače 0. Může být nulován též programově, avšak potom se ruší žádost o přerušení od čítače/časovače 0.

TR0 je start/stop čítače/časovače 0. Ve stavu log. 1 čítač 0 čítá. Bit se nastavuje a nuluje pouze programově.

IE1 je příznak žádosti o přerušení na vstupu INT1. Stav IE1 = 1 se nastaví vždy při příchodu signálu přerušení. Nuluje se automaticky při přechodu na adresu ošetření žádosti o přerušení od vnějšího vstupu INT1. Pokud je tento bit vynulován programově, zůstane v nule, pokud netrvá úrovňový požadavek na přerušení.

IT1 určuje druh reakce čítače 1. Stav IT1 = 1 znamená reakci na sestupnou hranu na vstupu INT1, stav IT1 = 0 znamená reakci na nulovou úroveň na vstupu INT1.

IE0 je příznak žádosti o přerušení na vstupu INT0. Stav IE1 = 1 se nastaví vždy při příchodu signálu přerušení.Nuluje se automaticky při přechodu na adresu ošetření žádosti o přerušení od vnějšího vstupu INT0. Pokud je tento bit vynulován programově, zůstane v nule, pokud netrvá úrovňový požadavek na přerušení.

IT0 určuje druh reakce čítače 0. Stav IT1 = 1 znamená reakci na sestupnou hranu na vstupu INT0, stav IT1 = 0 znamená reakci na nulovou úroveň na vstupu INT0.

Čítač/časovač 2

Obvod 80C32 obsahuje navíc šestnáctibitový počítací registr a šestnáctibitovou vyrovnávací paměť. Podle nastavení řídicího bitu CP/RL2 v registru T2CON lze vyrovnávací paměť využít buď pro obnovu obsahu počítacího registru, nebo lze do této vyrovnávací paměti vzorkovat obsah počítacího registru.

Další funkcí čítače/časovače 2 je generovat přenosové rychlosti pro sériový kanál.

Režim čítače/časovače 2 je určen registrem T2CON, jehož bity mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

kde:

TF2 je příznak přeplnění počítacího registru. Musí být nulován programově. Příznak TF2

není ovlivněn, pokud je čítač/časovač 2 využit pro řízení rychlosti přenosu sériového kanálu.

EXF2 je příznak požadavku externího přerušení. Přerušení je požadováno stavem log. 1, pokud je nastaven opakovací nebo vzorkovací režim a je povoleno externí přerušení bitem EXEN2 = 1. Při povoleném přerušení od čítače/ /časovače 2 vyvolá závěrná hrana na vstupu T2EX (port P1.1) podprogram pro obsluhu přerušení od čítače/časovače 2. Bit EXF2 musí být nulován programově.

RCLK je bit, který definuje zdroj přenosové rychlosti přijímače sériového kanálu. Pokud je ve stavu log. 1, je čítač/časovač 2 připojen na hodinový vstup přijímače sériového kanálu. Potom je každým přeplněním počítacího registru generován právě jeden hodinový impuls. Pokud je bit RCLK ve stavu log. 0, je čítač/časovač 2 připojen na hodinový vstup přijímače sériového kanálu výstup čítače/časovače 1. Pomocí čítačů/časovačů lze ovlivnit přenosovou rychlost jen pro režim 1 a 3 sériového kanálu. Vhodným propojením čítačů/časovačů 1 a 2 lze dosáhnout různých přenosových rychlostí přijímače sériového kanálu.

TCLK je bit, který definuje zdroj přenosové rychlosti vysílače sériového kanálu. Pokud je ve stavu log. 1, je čítač/časovač 2 připojen na hodinový vstup vysílače sériového kanálu. Potom je každým přeplněním počítacího registru generován právě jeden hodinový impuls. Pokud je bit TCLK ve stavu log. 0, je čítač/časovač 2 připojen na hodinový vstup přijímače sériového kanálu výstup čítače/časovače 1. Pomocí čítačů/časovačů lze ovlivnit přenosovou rychlost jen pro režim 1 a 3 sériového kanálu. Vhodným propojením čítačů/časovačů 1 a 2 lze dosáhnout různých přenosových rychlostí vysílače sériového kanálu.

EXEN2 je bit, který povoluje vnější přerušení na vstupu T2EX (port P1.1). Ve stavu log. 1 je vzorkován obsah počítacího registru nebo se obnovuje jeho obsah se závěrnou hranou signálu T2EX. To však pouze v případě, že se čítač/časovač 2 nepoužívá pro generování přenosové rychlosti sériového kanálu. Ve stavu log. 0 neovlivňuje signál T2EX chování počítacího registru.

TR2 je spouštěcí a zastavovací bit, kterým se ovládá chod čítače/časovače 2. Ve stavu log. 1 čítač/časovač 2 čítá, ve stavu log. 0 je zastaven a obsah počítacího registru se nemění. Bit TR2 se nastavuje a nuluje pouze programem.

C/T2 je řídicí bit, kterým se volí režim čítání nebo časování. Je-li bit ve stavu log. 1, pracuje čítač/časovač 2 jako čítač vnějších událostí na vstupu T2 (port P1.0). Ve stavu log. 0 čítač/časovač 2 čítá 1/12 kmitočtu oscilátoru. Obsah počítacího registru se v obou případech mění závěrnou hranou hodinových impulsů.

CP/RL2 je řídicí bit, kterým se volí režim opakování nebo vzorkování. Je-li bit ve stavu log. 0, je vždy obnoven obsah počítacího registru buď vnějším signálem T2EX nebo přeplněním. Je-li bit ve stavu log. 1, je nastaven režim vzorkování a vnějším signálem T2EX může být kdykoliv přepsán obsah počítacího registru do vyrovnávací paměti. Pokud je alespoň RCLK = 1 nebo TCLK = 1, neovlivňuje tento řídicí bit CP/ /RL2 obsah počítacího registru a čítač/ /časovač 2 pracuje v režimu opakování předvolby při přeplnění počítacího registru, tvořeného registry TH2 a TL2.

Opakovací režim čítače/časovače 2

Při zapojení čítače/časovače 2 v opakovacím režimu se musí předvolená hodnota zapsat do dvou registrů předvolby RCAP2H a RCAP2L. Dva registry jsou použity proto, aby dohromady tvořily šestnáctibitový registr předvolby. Vyšší byte se zapisuje do RCAP2H, nižší do RCAP2L. Při každém vynulování obsahu počítacího registru, který je tvořen dvojicí osmibitových registrů TH2 a TL2, se tyto registry naplní obsahem registrů RCAP2H a RCAP2L. Pokud není povoleno řízení vnějším signálem T2EX (EXEN2 = 0), obnoví se obsah počítacího registru jen při jeho přeplnění. Při povoleném řízení vnějším signálem T2EX, a pokud je EXEN2 = 1, obnovuje se obsah počítacího registru kromě okamžiku přeplnění ještě tehdy, když je zjištěna sestupná hrana vstupního signálu T2EX. Touto sestupnou hranou je nastaven příznak žádosti o externí přerušení stejně, jako se přeplněním počítacího registru nastavuje příznak žádosti o přerušení od čítače/časovače 2.

Vzorkovací režim čítače/časovače 2

V tomto režimu se přepisuje obsah počítacího registru TH2 a TL2 do vyrovnávacího registru RCAP2H a RCAP2L závěrnou hranou externího signálu T2EX, pokud je přepis povolen řídicím bitem EXEN2 = 1. Současně s přepisem se nastaví příznak žádosti o přerušení od externího zdroje T2EX. Žádost o přerušení, vzniklá přeplněním počítacího registru čítače/časovače 2, je ošetřována na stejné adrese, jako žádost EXF2 od vnějšího zdroje žádosti T2EX.

Čítač/časovač 2 pro generování přenosové rychlosti sériového kanálu

Pokud je vybrán čítač/časovač 2 alespoň pro vysílač nebo přijímač jako zdroj hodinových impulsů, je zrušeno nastavení jeho původního režimu a čítač/časovač 2 generuje pouze vnitřní periodický signál.

Sériový kanál

Sériový kanál je plně duplexní (full-duplex), vysílač a přijímač tedy mohou pracovat na sobě nezávisle. Přijímač i vysílač jsou vybaveny vyrovnávací pamětí pro jeden znak. Obě vyrovnávací paměti jsou adresovány jedinou adresou SFR s názvem SBUF.

Zápisem do SBUF se plní vysílač, čtením SBUF se odebírá slovo z přijímače. Přeplnění vysílače a přetečení přijímače se musí vyloučit programově.

Sériový kanál se řídí registrem s názvem SCON, jehož bity mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

kde:

SM0 a SM1 jsou bity, které nastavují režim činnosti sériového kanálu - viz tab. 13.

SM2 je bit, který povoluje multiprocesorové komunikace v režimech 2 a 3. Pokud je SM2 ve stavu log. 1, RI se aktivuje jednotkovým devátým bitem. V režimu 1 se RI aktivuje správným stop bitem (log. 1). Pro režim 0 by SM2 měl být ve stavu log. 0. V režimech 2 a 3 se devátý bit ukládá do bitu RB8.

REN je bit, který povoluje příjem sériového kanálu. Nastavuje a nuluje se programově.

TB8 definuje hodnotu devátého bitu pro vysílač a je použit jako devátý bit vysílaného znaku v režimech 2 a 3. Nastavuje se programově.

RB8 je bit, který se určí příjmem devátého bitu. V režimu 1 odpovídá přijatému bitu STOP, v režimech 2 a 3 je to skutečný devátý datový bit. V režimu 0 je bez významu. Čtením není nulován.

TI je příznakem žádosti o přerušení od vysílače. Nastavuje se automaticky po osmém bitu v režimu 0, v režimech 1 až 3 při začátku vysílání bitu STOP. Musí být nulován programově, jinak není možno rozpoznat, že vysílač je plný.

RI je příznakem žádosti o přerušení od přijímače. Nastavuje se automaticky po příjmu osmého bitu v režimu 0 nebo po příjmu první poloviny bitu STOP v ostatních režimech. Výjimku tvoří při nastaveném bitu SM2 do stavu log. 1 režim 2 a 3. Pokud nebude devátý bit ve stavu log. 1, nenastaví se při bitu SM2 ve stavu log. 1 bit RI do stavu log. 1 při příjmu druhé poloviny bitu STOP. Příznak RI musí být nulován programově, jinak není možné rozpoznat, že přijímač je připraven vydat přijmutý znak. RI musí být programově nulován po přečtení znaku z přijímače.

Ovládání sériového kanálu se liší v režimu 0 od ostatních režimů. Povolení příjmu sériového slova je vázáno na splnění dvou podmínek. Bit RI musí být ve stavu log. 0 a současně musí být povolen příjem bitem REN ve stavu log. 1. V ostatních režimech záleží pouze na bitu REN, zda bude nebo nebude pracovat přijímač.

Režim 0

Sériová data vstupují a vystupují přes linku RxD. Na lince TxD vystupují synchronizační impulsy. Na straně přijímače je linka TxD využita jako vstup hodinových impulsů s přenosovou rychlostí. Protože datové slovo tvoří 8 bitů, je vysíláno taktéž 8 synchronizačních impulsů. Jako první se vysílá LSB (nejnižší bit). Nejvyšší přenosová rychlost může být max. 1/12 kmitočtu oscilátoru. Závěrné hrany generovaných synchronizačních impulsů určují platnost datových bitů na lince RxD. Data zůstávají platná po dobu TxD = 0. Režim 0 není duplexní.

Režim 1

Tento režim je určen pro přenos desetibitových slov s délkou dat osm bitů. Slova jsou vysílána na lince TxD a v duplexním režimu přijímána na lince RxD. Každé osmibitové datové slovo je doplněno na začátku bitem START a na konci bitem STOP. Při příjmu se bit START nikam nezapisuje, osm datových bitů se zapíše do registru SBUS a bit STOP se zapíše do stavového bitu RB8 v registru SCON. Slovo se vysílá v pořadí bit START, datové slovo s prvním bitem LSB a nakonec se vysílá bit STOP.

Přenosová rychlost je určena přeplněním počítacího registru čítače/časovače 1 nebo 2. Vzhledem k vnitřnímu děliči 1 : 16 přenosové rychlosti je vnitřní časování kanálu přijímače a vysílače 16x rychlejší. Datová linka se čte v sedmém, osmém a devátém hodinovém impulsu.

Režim 2

Tento režim je určen pro přenos jedenáctibitových slov. Kromě bitu START a STOP je datové slovo doplněno devátým bitem, jehož hodnotu lze volit zápisem příslušné hodnoty do bitu TB8 v registru SCON. Pokud bude do TB8 zapsána hodnota parity osmibitového datového slova, budou se vysílat a přijímat datová slova s kontrolním (paritním) bitem. Při příjmu se bity START a STOP nikam nezapisují, osm nižších datových bitů se zapíše do registru SBUF pro následné čtení a devátý bit se přenese do bitu RB8 registru SCON. Rychlost přenosu lze volit ze dvou přenosových rychlostí podle nastavení bitu SMOD v registru PCON. Tímto bitem lze nastavit buď 1/32 nebo 1/64 kmitočtu oscilátoru. Vzorkování dat v přijímači je totožné s režimem 1.

Režim 3

Režim 3 je podobný režimu 2. Jsou stejně přenášena jedenáctibitová slova s doplněným devátým bitem v datové části. Rozdíl je v pře-

Tab. 13. Význam bitů SM0 a SM1

SM0	SM1	Režim	Činnost
0	0	0	synchronní
0	1	1	osmibitový UART
1	0	2	devítibitový UART
1	1	3	devítibitový UART

nosové rychlosti. V režimu 3 je volitelná přenosová rychlost a může být generována přetečením čítače/časovače 1 nebo 2. Vzorkování dat v přijímači je totožné s režimem 1.

Mikropočítačové komunikace

Stavový bit SM2 registru SMOD určuje podmínky, za kterých může sériový kanál přijímače žádat o přerušení. Je-li SM2 ve stavu log.1, je v režimech 1, 2 a 3 žádáno přerušení při sériovém příjmu jen tehdy, když devátý bit bude přijatý jako log. 1. Provádí se test obsahu stavového bitu RB8. Bit RB8 ve stavu log. 0 potlačí žádost o přerušení od přijímače. Pokud sériový kanál pracuje v režimu 1, bude testován přímo bit STOP. V režimech 2 a 3 obsahuje stavový bit RB8 vždy hodnotu devátého bitu.

Mikropočítačové sítě

Mikropočítače, zapojené do sítě, pracují v konfiguracích MASTER a SLAVE. MASTER rozesílá příslušná data jednotlivým mikropočítačům SLAVE.

Každý mikropočítač SLAVE nastaví režim svého sériového kanálu na 2 nebo 3 a nastaví SM2 do stavu log. 1. Mikropočítač MASTER uloží do prvního vysílaného slova adresu přijímacího mikropočítače SLAVE a devátý bit nastaví na log. 1 pomocí TB8. MASTER vyšle slovo, kterým se naplní všechny mikropočítače SLAVE. Protože v přijímačích je TB8 ve stavu log. 1 a byl nastaven SM2 do stavu log. 1, všechny mikropočítače SLAVE vykonají přerušení od přijímače. V podprogramu, který ošetřuje žádost o přerušení od přijímače, porovnává každý mikropočítač SLAVE vlastní programově přidělenou adresu s vyslanou adresou, získanou přijatým datovým osmibitovým slovem. Při shodě adres přejde mikropočítač SLAVE do programu, který testuje stavový bit RI a odebírá přicházející sériové znaky.

V rámci tohoto programu je nutné vynulovat řídicí bit režimu SM2. Tím je nastavena podmínka pro generování žádosti o přerušení od přijímače po každém přijatém slově bez ohledu na stav devátého bitu. Ostatní mikropočítače SLAVE se vracejí do předchozího programu a čekají na další žádost o přerušení od přijímače.

Mikropočítač MASTER po vyslání adresy pokračuje ve vysílání dalších slov, ve kterých je však vždy TB8 ve stavu log. 0, aby nebylo vynucováno přerušení ostatních mikropočítačů SLAVE. Neadresované mikropočítače SLAVE pak tento datový blok ignorují a čekají na adresové slovo, kde RB8 bude va stavu log. 1. Po ukončení přenosu musí adresovatelné mikropočítače SLAVE přejít znovu do režimu s nastaveným SM2 do stavu log. 1, aby mohly znovu žádat o přerušení s příchodem adresového slova.

Nastavení přenosové rychlosti čítačem/časovačem 1

Přenosová rychlost sériového kanálu je určena v různých režimech následujícími vzorci:

Režim 0: (kmitočet oscilátoru) / 12.

Režim 1 a 3: $2^{SMOD}/32$ x (přeplnění čítače 1).
nebo
$[2^{SMOD}/32]$ x [kmitočet oscilátoru / / [12 x (256 – TH1)]].

Režim 2: $2^{SMOD}/64$ x (kmitočet oscilátoru).

Standardní přenosové rychlosti jsou generovány časovačem 1 za podmínek, které jsou uvedené v tab. 14.

U mikropočítačů typu 80C32 a 80C52 lze využít ke generování přenosové rychlosti sériového kanálu také čítač/časovač 2. Nastavením bitu TCLK nebo RCLK registru T2CON budou vysílač nebo přijímač nebo vysílač i přijímač řízeny přímo výstupem čítače/časovače 2. Přenosová rychlost bude potom určena vztahem:

Režim 1 a 3: (kmitočet oscilátoru) / / 32 x [65536 – (RCAP2H, RCAP2L)].

Inicializace po zapnutí napájecího napětí a RESET

Na obr. 19 je schéma nulovacího obvodu, který po připojení napájecího napětí k mikropočítači automaticky generuje signál RESET.

Inicializace se vždy provede funkcí RESET. Po zapnutí napájecího napětí vykoná funkci RESET člen *RC*, který je připojen na vstup RST mikropočítače, funkce však může být vykonána i přidržením úrovně log. 1 na vstup RST po určitou dobu. Vstup RST je vybaven úrovňovým komparátorem, a proto je možné připojit bez dalšího ošetření i spojitě se měnící analogový signál.

Inicializace mikropočítače proběhne bez chyby pouze v tom případě, pokud je signál RST udržen v úrovni log. 1 alespoň dva strojové cykly (24 period kmitočtu oscilátoru) a za předpokladu, že oscilátor běží. Pokud se oscilátor rovněž rozbíhá (po připojení napájecího napětí), je doporučeno přidržet úroveň log. 1 na vstupu RST alespoň po dobu 1 ms.

Během inicializace přejdou do neaktivního stavu signály ALE a PSEN (non), během druhého strojního cyklu probíhá inicializace obsahu registrů SFR. Pokud signál RST trvá déle, opakují se tyto inicializační sekvence až do okamžiku ukončení signálu RST.

Při inicializaci se nemění obsah vnitřní paměti dat. Po připojení napájecího napětí je stav obsahu vnitřní paměti dat náhodný. Výjimku tvoří pouze návrat z režimu se sníženým příkonem, kdy je v paměti RAM uchován původní obsah (nejde totiž o obnovu napájecího napětí, ale pouze o přechod z udržovacího napájení na plný výkon).

V tab. 15 je uveden stav registrů SFR po inicializaci mikropočítače.

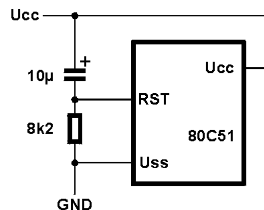
Režim sníženého příkonu

Mikropočítače řady „51" mohou být uvedeny do režimu sníženého příkonu.

U mikropočítačů typu HMOS je možné přejít do režimu sníženého příkonu generováním

Tab. 14. Generování standardních přenosových rychlostí časovačem 1

Přenosová rychlost		f_{osc}	SMOD	C/T	Režim časovače	Hodnota předvolby
Režim 0:	1 MHz	12 MHz	X	X	X	X
Režim 2:	375 kHz	12 MHz	1	X	X	X
	178,5 kHz	12 MHz	0	X	X	X
Režim 1, 3:	62,5 kHz	11,059 MHz	1	0	2	0FFH
	19,2 kHz	11,059 MHz	1	0	2	0FDH
	9,6 kHz	11,059 MHz	0	0	2	0FDH
	4,8 kHz	11,059 MHz	0	0	2	0FAH
	2,4 kHz	11,059 MHz	0	0	2	0F4H
	1,2 kHz	11,059 MHz	0	0	2	0E8H
	0,11 kHz	6 MHz	0	0	2	072H
	0,11 kHz	12 MHz	0	0	1	0FEEBH



Obr. 19. Nulovací obvod

Tab. 15. Stav speciálních funkčních registrů SFR po inicializaci mikropočítače

Registr	Obsah	Registr	Obsah
Acc	00H	IP	XX000000B
B	00H	IE	0X000000B
PSW	00H	PCON	0XXX0000B
SP	07H	TMOD	00H
P0 až P3	0FFH	TH0	00H
PC	0000H	TH1	00H
DPTR	0000H	TL0	00H
T2CON	00H	TL1	00H
TH2	00H	TCON	00H
TL2	00H	SCON	00H
RCAP2H	00H	SBUF	nedefinován
RCAP2L	00H		

signálu RST. Pokud je udrženo na vývodu RST napětí alespoň 2,4 V, postačí toto napětí na udržení obsahu vnitřní RAM. Po nastavení RST do úrovně log. 1 a po prodlevě dvou strojních cyklů je možné zcela odpojit napájecí napětí *Ucc* mikropočítače. Při přechodu zpět se obnoví napájecí napětí a signál RST se zruší. Mikropočítač bude dále pokračovat v činnosti. Pokud je nutné zajistit pokračování programu před a po odpojení napájecího napětí, musí se před přechodem do režimu sníženého příkonu uschovat registry SFR do RAM a po návratu programově opět registry SFR naplnit původním obsahem.

U obvodů typu CMOS může být příkon snížen dvěma způsoby. První z nich je IDLE MODE a druhý je POWER DOWN MOD.

Režim IDLE MODE je charakteristický tím, že pracuje přerušovací systém a hodinový generátor, který je připojen k obvodům čítačů a sériového kanálu. Stav mikropočítače je zachován, jsou zachovány obsahy registrů SFR a obsah celé interní paměti RAM. Vývody datových bran zůstávají v posledních nastavených úrovních. Signály ALE a PSEN (non) jsou neaktivní. IDLE MODE se nastavuje řídicím bitem IDL v registru PCON. Po nastavení bitu IDL do stavu log. 1 přejde v následujícím strojním cyklu mikropočítač do režimu IDLE MODE. Obnovení činnosti je možné generováním některé z povolených žádostí o přerušení. Obsahy registrů SFR a vnitřní RAM se nemění. Když se dokončí podprogram, který ošetří žádost o přerušení, a provede se instrukce RETI, pokračuje program za instrukcí, která přechod do režimu IDLE MODE vyvolala. Druhým možným způsobem ukončení režimu IDLE MODE je aktivace signálu RST po dobu dvou strojních cyklů. Příznakové bity GF1 a GF0 mohou sloužit k označení typu režimu i způsobu jeho opuštění.

Bity registru PCON mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	-	-	-	GF1	GF0	PD	IDL

kde:

SMOD je řídicí bit děliče přenosové rychlosti. Pro SMOD ve stavu log. 0 je přenosová rychlost poloviční.

GF1 je příznak pro rozlišení návratu z režimů s potlačeným příkonem.

GF0 je příznak pro rozlišení návratu z režimů s potlačeným příkonem.

PD je řídicí bit pro přechod na snížené napájení. Po nastavení na bitu do stavu log. 1 se

v příštím strojním cyklu potlačí funkce mikropočítače. Vrátit se lze pouze inicializací.

IDL je řídicí bit pro přechod do režimu s potlačenou činností. Po nastavení bitu do stavu log. 1 se v příštím strojním cyklu potlačí funkce mikropočítače. Ve funkci zůstávají čítače/časovače, sériový kanál a přerušovací systém. Vrátit se lze buď generováním povolené žádosti o přerušení nebo inicializací.

Režim se sníženým příkonem POWER DOWN MOD je významnou funkcí, která skutečně značně omezí příkon. Do tohoto režimu je možné přejít po instrukci, která v registru PCON nastaví bit PD do stavu log. 1. V tomto stavu se zastavuje oscilátor i tím i chod celého mikropočítače. Stav mikropočítače je zachován, jsou zachovány obsahy registrů SFR a obsah celé interní paměti RAM. Vývody datových bran zůstávají v posledních nastavených úrovních. Čítače, sériový kanál ani přerušovací systém nepracují. Výstup z tohoto režimu je možný pouze inicializací mikropočítače uvedením signálu RST do úrovně log. 1. Protože však nepracoval oscilátor, je nutné udržet stav log. 1 na vstupu RST po dobu alespoň 1 ms. Napájecí napětí lze snížit až po nastavení bitu PD1 do stavu log. 1 a po odeznění jednoho strojního cyklu. Před generováním signálu RST musí být obnoveno napájecí napětí na jmenovitou velikost. Ochrana registrů SFR se může zajistit pouze prostřednictvím vnitřní RAM, která není ovlivněna změnou provozního režimu.

Oscilátor mikropočítače

Zapojení oscilátoru mikropočítače s krystalem a s vnějším taktovacím signálem je na obr. 20 a obr. 21.

Hodinový generátor uvnitř mikropočítače obsahuje Piercův oscilátor. Kmitočet oscilátoru se může pohybovat v rozmezí, které udává výrobce daného typu mikropočítače, většinou je to v rozmezí 4 až 20 MHz. Paralelní kapacity je vhodné volit 27 pF pro kvalitní krystal, pro keramické rezonátory jsou potřebné kapacity 47 pF.

Taktovací signál z vnějšího oscilátoru lze u mikropočítače, vyrobeného technologií CMOS, připojit na vstup XTAL1, přičemž vstup XTAL2 se nezapojí.

U technologií HMOS nebo NMOS se vnější oscilátor připojí na vstup XTAL2 a vstup XTAL1 se uzemní.

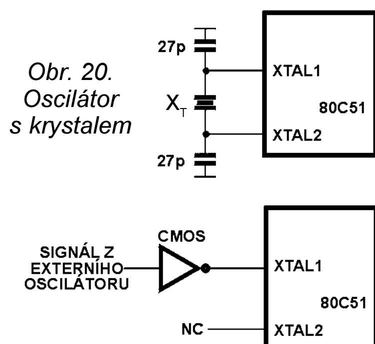
Odlaďování programu

Pro odlaďováni programů můžeme použít programové simulátory, simulátory EPROM nebo v nejlepším případě použijeme emulátor.

Programové simulátory jsou programy na PC, které umožňují bez spojení s hardwarem vývojové aplikace po přeložení zdrojového programu vypisovat na obrazovku monitoru PC obsahy registrů mikropočítače. Program lze spustit, krokovat atd. Z hlediska vývojáře se tento způsob jeví nešťastným, doporučuje se však pro úplné začátečníky k ověření funkce mikropočítače.

Simulátory EPROM simulují programovou paměť mikropočítače. Po napsání celku nebo části programu a po jeho přeložení do strojového kódu asemblerem „51" se výsledek přeloží do formátu HEX, který se vhodným programátorem EPROM naprogramuje do paměti simulátoru. Vlastní simulátor je většinou tvořen pamětí typu SRAM (statická RAM). Naprogramovaná paměť se potom přenese do aplikace. Tento způsob je již celkem dokonalý, přináší však nevýhodu v náročnosti na čas a v manipulaci se zařízením.

Emulátory jsou nejdokonalejším vývojovým prostředkem, s jehož pomocí je přímo spojen počítač PC s vývojovou aplikací. Překlad programu a ovládání aplikace jsou většinou zautomatizovány tak, že po napsání programu se vhodným příkazem přeloží zdrojový program do strojového kódu a aplikace je spuštěna v reálném čase.



Obr. 20. Oscilátor s krystalem

Obr. 21. Připojení vnějšího taktovacího signálu k mikropočítači

Popis mikropočítače 80C552

Mikropočítač 80C552 je osmibitový jednočipový mikrořadič, který se používá pro náročnější aplikace. Mikropočítač plně zahrnuje instrukční soubor řady „51", oproti typu 80C51 má však rozšířen počet registrů SFR, které odpovídají periferiím na čipu. Mikropočítač je vyráběn technologií CMOS. Typ 83C552 obsahuje 8 kbyte paměti ROM a 256 byte RAM, typ 87C552 obsahuje 8 kbyte paměti EPROM a 256 byte RAM a typ 80C552 neobsahuje vnitřní paměť programu a má 256 byte RAM.

Mikropočítač řady „552" obsahuje pět osmibitových vstupně/výstupních datových bran a jednu osmibitovou datovou vstupní bránu, dva šestnáctibitové čítače/časovače a jeden šestnáctibitový časovač se záchytnou (střádací) a porovnávací vyrovnávací pamětí, přerušovací strukturu, osmivstupový desetibitový analogově/digitální (A/D) převodník ADC s multiplexem, dvojitý digitálně/analogový převodník DAC s rozhraním se šířkově modulovanými impulsy, dvě sériová rozhraní UART a I2CBUS, obvod pro hlídání chodu programu WATCHDOG, hodinový oscilátor a časovací logiku. Typy 80x552 mají dva softwarově vybíratelné mody pro řízení spotřeby IDDLE a POWER DOWN.

Sériový port SIO1 (I2C) má standardní funkce tohoto rozhraní a může být připojen ke všem obvodům, které toto rozhraní obsahují.

Mikropočítač obsahuje pět osmibitových vstupně/výstupních datových bran.

Porty P1.6 a P1.7 mají otevřené kolektorové výstupy pro spolupráci s I2C, a pokud budou používány jako standardní porty, musí být doplněny rezistory (pull-up), připojenými na napájecí sběrnici. Zapojení vývodů pouzdra mikropočítače 80C552 je obr. 22.

Paměť programu

U typů 87C552 a 83C552 má paměť programu na čipu velikosti 8 kbyte a může být doplněna o externí paměť až do velikosti 64 kbyte. Pokud je na vývod EA (non) přivedena vysoká úroveň (napájecí napětí), čte mikropočítač program ze své interní programové paměti až do adresy 1FFFH. Od adresy 2000H do adresy 0FFFFH čte programovou paměť externí. Pokud je na vývod EA (non) přivedena nízká úroveň, veškeré instrukce jsou čteny z externí programové paměti. Adresy od 0003H do 0073H jsou rezervovány pro přerušovací logiku.

Paměť dat

Vnitřní paměť dat mikropočítače je rozdělena do tří sekcí. První obsahuje nejnižších 128 byte RAM, druhá vyšších 128 byte RAM a třetí 128 byte pro speciální funkční registry SFR. Nejnižších 128 byte je přímo a nepřímo adresovatelných. Pokud je naadresováno od 128 do 255 nebo jsou naadresovány speciální funkční registry SFR, musí být adresováno rozdílnými adresovacími módy. RAM 128 až 255 je adresovatelná pouze nepřímo a speciální funkční registry jsou adresovatelné pouze přímo. Adresuje se stejným způsobem, jako u typu 80C51.

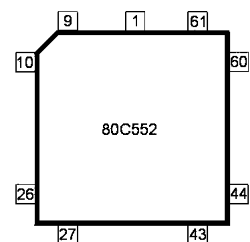
Zásobníková paměť může být adresována adresovým ukazovátkem (stack pointer), které má osm bitů, takže ukazovátko může adresovat max. 256 byte.

Speciální funkční registry SFR

Speciální funkční registry jsou pouze přímo adresovatelné kromě programového čítače a čtyř bank registrů R. Většina z padesáti šesti speciálních funkčních registrů jsou použity pro řízení jednotlivých hardwarových bloků na čipu. Některé registry zahrnují aritmeticko/logické registry (Acc.B,PSW) a datový ukazatel registrů (DPH a DPL které tvoří registrový pár DPTR). Šestnáct z registrů SFR může být bitově adresováno. Adresy a popis speciálních funkčních registrů je v tab. 16.

Časovač T2

Časovač T2 je šestnáctibitový čítač/časovač, složený ze dvou registrových párů TML2 (dolní byte) a TMH2 (horní byte). Na vstup tohoto šestnáctibitového čítače/časovače mohou být přivedeny hodinové impulsy z předděličky 1/12 kmitočtu oscilátoru nebo vstup může být připojen na externí signál. Pokud je časovač T2 konfigurován jako čítač, předdělička je ovládána hodinovým signálem, který se přivádí na vstup T2 (P1.4). Předdělička je ovládána ná-



1. P5.0/ADC0	35. XTAL1
2. Ucc	36. Uss
3. STADC	37. Uss
4. PWM0	38. NC
5. PWM1	39. P2.0/A08
6. EW	40. P2.1/A09
7. P4.0/CMSR0	41. P2.2/A10
8. P4.1/CMSR1	42. P2.3/A11
9. P4.2/CMSR2	43. P2.4/A12
10. P4.3/CMSR3	44. P2.5/A13
11. P4.4/CMSR4	45. P2.6/A14
12. P4.5/CMSR5	46. P2.7/A15
13. P4.6/CMT0	47. PSEN
14. P4.7/CMT1	48. ALE
15. RST	49. EA
16. P1.0/CT0I	50. P0.7/AD7
17. P1:1/CT1I	51. P0.6/AD6
18. P1.2/CT2I	52. P0.5/AD5
19. P1.3/CT3I	53. P0.4/AD4
20. P1.4/T2	54. P0.3/AD3
21. P1.5/RT2	55. P0.2/AD2
22. P1.6/SCL	56. P0.1/AD1
23. P1.7/SDA	57. P0.0/AD0
24. P3.0/RxD	58. AV ref-
25. P3.1/TxD	59. AV ref+
26. P3.2/INT0	60. AV ss
27. P3.3/INT1	61. AV dd
28. P3.4/T0	62. P5.7/ADC7
29. P3.5/T1	63. P5.6/ADC6
30. P3.6/WR	64. P5.5/ADC5
31. P3.7/RD	65. P5.4/ADC4
32. NC	66. P5.3/ADC3
33. NC	67. P5.2/ADC2
34. XTAL2	68. P5.1/ADC1

Obr. 22. Zapojení vývodů pouzdra mikropočítače 80C552

stupnou hranou vstupního signálu, jehož maximální kmitočet je 1/12 kmitočtu krystalu (1 MHz při použití krystalu 12 MHz). Náběžná hrana vstupního signálu je detekována, pokud T2 je na nízké úrovni během prvního vzorku a na vysoké úrovni během vzorku následujícího. Předdělička je programovatelná pro dělení čísly 1, 2, 4 a 8 a je nulována, pokud je změněn dělicí poměr nebo je změněn vstupní zdroj. Taktéž je nulována, pokud je čítač/časovač 2 nulován.

Čítač T2 nelze plnit, pokud je mikropočítač ve stavu RESET. Nulování čítače T2 vnějším signálem na vývodu RT2 se uvolňuje nastavením bitu T2ER v registru TM2CON. Přetečení čítače může vyvolat přerušení, přerušení se uvolňuje softwarově v registru IEN1.

Bity registru IEN1 mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0

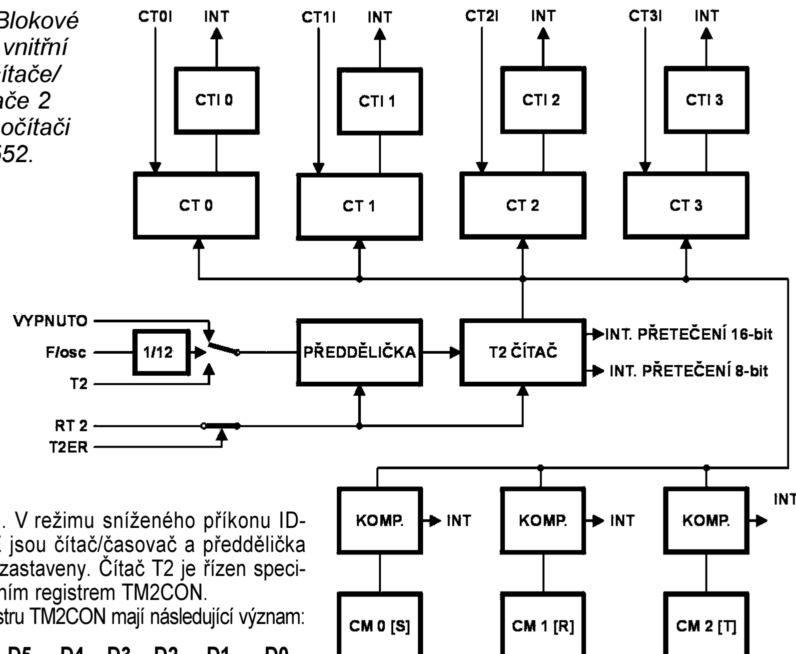
kde:

ET2 uvolňuje přerušení při přetečení časovače T1.
ECM2 uvolňuje přerušení od komparátoru 2.
ECM1 uvolňuje přerušení od komparátoru 1.
ECM0 uvolňuje přerušení od komparátoru 0.
ECT3 uvolňuje přerušení od střadače 3.
ECT2 uvolňuje přerušení od střadače 2.
ECT1 uvolňuje přerušení od střadače 1.
ECT0 uvolňuje přerušení od střadače 0.

Pokud je vyžadováno přerušení při přetečení všech šestnácti bitů čítače, musí být nastaveny bity ET2 (celkové uvolnění přerušení) a T2IS1 (výběr přerušení při šestnáctibitovém přetečení). Bit T2OV je příznak (flag) žádosti o přerušení při přetečení čítače. Všechny příznaky žádostí o přerušení musí být nulovány softwarově. Pokud je vyžadováno přerušení při přetečení byte i při šestnáctibitovém přetečení, musí být nastaveny bity T2IS0 a T2IS1 a musí být použity dva podprogramy pro obsluhu přerušení. V každém podprogramu musí být nulován příznak přetečení.

Čítač T2 může být nulován náběžnou hranou signálu na vnějším vstupu RT2 (P1.5), pokud je povoleno vnější nulování nastavením bitu T2ER. Vnější nulovací signál nuluje taktéž předděličku. V režimu sníženého příkonu IDLE MODE jsou čítač/časovač a předdělička nulovány a zastaveny. Čítač T2 je řízen speciálním funkčním registrem TM2CON.

Obr. 23. Blokové schéma vnitřní logiky čítače/časovače 2 v mikropočítači 80C552.



Bity registru TM2CON mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
T2IS1	T2IS0	T2ER	T2BO	T2P1	T2P0	T2MS1	T2MS0

kde:

T2IS1 je výběr přerušení při šestnáctibit. přetečení.

T2IS0 je výběr přerušení při osmibit. přetečení.

T2ER je uvolnění vnějšího nulování čítače T2. Pokud je tento bit nastaven do stavu log. 1, čítač T2 může být nulován vzestupnou hranou signálu na vývodu RT2 (P1.5).

T2BO je příznak žádosti o přerušení při přetečení.

T2P1 je nastavení předděličky (bit 1) - viz tab. 17.

T2P0 je nastavení předděličky (bit 0) - viz tab. 17.

T2MS1 je výběr modu čítače T2 (bit 1) - viz tab. 18.

T2MS0 je výběr modu čítače T2 (bit 0) - viz tab. 18.

Pokud je použit krystal 12 MHz, přeteče časovač v šestnácti bitech vždy po 65,5 ms, 131 ms, 262 ms a nebo 524 ms, což záleží na nastavení dělicího poměru v předděličce.

Střádací a porovnávací logika čítače 2

Čítač T2 je připojen uvnitř mikropočítače na čtyři šestnáctibitové střádací registry a na tři registry porovnávací. Střádací registry mohou sloužit k běžnému zachytávání hodnot čítače, porovnávací registry pro nastavení a nulování výstupních linek brány 4 a také k předprogramování časových intervalů. Kombinací střádacích a porovnávacích registrů je možné sestrojit velice mocný čítač.

Střádací logika čítače 2

Na obr. 23 je blokové schéma vnitřní logiky čítače/časovače 2.

CT0, CT1, CT2 a CT3 jsou čtyři střádací registry, které jsou připojeny na čítač T2. Bylo by lépe používat termín záchytné registry, protože

Tab. 16. Speciální funkční registry SFR mikropočítače 80C552

Adresa	Popis	Zkratka	Adresa	Popis	Zkratka
80H	port P0	P0	C8H	registr T2CON	T2CON
81H	ukazatel zásobníkové paměti	SP	C9H	porovnávací registr 0 - horní byte	CMH0
82H	registr DPTR - dolní byte	DPL	CAH	porovnávací registr 1 - horní byte	CMH1
83H	registr DPTR – horní byte	DPH	CBH	porovnávací registr 2 - horní byte	CMH2
87H	registr PCON	PCON	CCH	střadač 0 - horní byte	CTH0
88H	registr TCON	TCON	CDH	střadač 1 - horní byte	CTH1
89H	registr TMOD	TMOD	CEH	střadač 2 - horní byte	CTH2
8AH	počáteční hodnota čítače T0 - dolní byte	TL0	CFH	střadač 3 - horní byte	CTH3
8BH	počáteční hodnota čítače T1 - dolní byte	TL1	D0H	registr stavového slova	PSW
8CH	počáteční hodnota čítače T0 - horní byte	TH0	D8H	řízení sériového kanálu 1	SICON
8DH	počáteční hodnota čítače T1 - horní byte	TH1	D9H	stavové slovo sér. kanálu 1	S1STA
90H	port P1	P1	DAH	data sér. kanálu 1	SIDAT
98H	registr S0CON	S0CON	DBH	adresa sér. kanálu 1	SIADR
99H	registr sériového kanálu	S0BUF	E0H	akumulátor Acc	A
A0H	port P2	P2	E8H	uvolnění přerušení 1	IEN1
A8H	uvolnění přerušení 0	IEN0	EAH	řízení čítače 2	TM2CON
A9H	porovnávací registr 0 - dolní byte	CML0	EBH	řízení střadače	CTCON
AAH	porovnávací registr 1 - dolní byte	CML1	ECH	registr čítače 2 – dolní byte	TML2
ABH	porovnávací registr 2 - dolní byte	CML2	EDH	registr čítače 2 – horní byte	TMH2
ACH	střadač 0 - dolní byte	CTL0	EEH	nastavení uvolnění	STE
ADH	střadač 1 - dolní byte	CTL1	EFH	nulování/překlopení uvolnění	RTE
AEH	střadač 2 - dolní byte	CTL2	F0H	registr B	B
AFH	střadač 3 - dolní byte	CTL3	F8H	registr priority přerušení IP1	IP1
B0H	port P3	P3	FCH	registr PWM0	PWM0
B8H	registr priority přerušení IP0	IP0	FDH	registr PWM1	PWM1
C0H	port P4	P4	FEH	registr PWM přednastavení	PWMP
C5H	řízení ADC převodníku	ADCON	FFH	časovač T3	T3
C6H	A/D převodník - horní byte	ADCH			

Tab. 17. Nastavení předděličky

T2P1	T2P0	T2 hodiny
0	0	nedělený poměr
0	1	zdroj hodinového signálu/2
1	0	zdroj hodinového signálu/4
1	1	zdroj hodinového signálu/8

Tab. 18. Výběr modu čítače T2

T2MS1	T2MS0	MÓD
0	0	čítač T2 je zastaven
0	1	čítač T2 pracuje s 1/12 kmitočtu oscilátoru
1	0	testovací mod - nepoužito
1	1	čítač T2 pracuje s externím signálem, přivedeným na pin T2

vlastní střadač informací v mikropočítači je akumulátor Acc. Běžně se však pro obvod, uchovávající informace, používá termín střadač, takže v dalším textu bude tento termín používán.

Tyto střádací registry jsou plněny obsahem časovače T2 a mohou vyvolávat přerušení ve spolupráci s vnějšími přerušovacími signály CT0I, CT1I, CT2I a CT3I. Signály jsou přítomny na bráně (portu) 1. Pokud nebude využita střádací logika, mohou tyto vstupy sloužit jako další externí vstupy pro přerušení. Na přerušení upozorňují čtyři příznaky žádostí o přerušení v registru TM2IR.

Ve speciálním funkčním registru CTCON mohou vstupy reagovat na vzestupnou, sestupnou nebo na obě hrany signálu. Vstupy jsou vzorkovány během každého cyklu S1P1.

Bity registru CTCON mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
CTN3	CTP3	CTN2	CPT2	CTN1	CPT1	CTN0	CPT0

kde:

CTN3 je bit, při jehož stavu log. 1 je střadač 3 ovládán sestupnou hranou na vývodu CT3I.

CPT3 je bit, při jehož stavu log. 1 je střadač 3 ovládán vzestupnou hranou na vývodu CT3I.

CTN2 je bit, při jehož stavu log. 1 je střadač 2 je ovládán sestupnou hranou na vývodu CT2I.

CPT2 je bit, při jehož stavu log. 1 je střadač 2 je ovládán vzestupnou hranou na vývodu CT2I.

CTN1 je bit, při jehož stavu log. 1 je střadač 1 je ovládán sestupnou hranou na vývodu CT1I.

CPT1 je bit, při jehož stavu log. 1 je střadač 1 je ovládán vzestupnou hranou na vývodu CT1I.

CTN0 je bit, při jehož stavu log. 1 je střadač 0 je ovládán sestupnou hranou na vývodu CT0I.

CPT0 je bit, při jehož stavu log. 1 je střadač 0 je ovládán vzestupnou hranou na vývodu CT0I.

Měření časových intervalů pomocí střadačů

Pokud je periodický vnější signál reprezentován ve formě vzestupných a sestupných hran logických úrovní, lze tento signál připojit ke vstupu jednoho ze čtyř střadačů. Časové intervaly vnějšího signálu mohou být potom měřeny pomocí čítače T2 a připojeného střadače. Pokud nastane vnější událost (nakonfigurovaná příslušná hrana signálu), je obsah časovače T2 okamžitě kopírován do střadače a je generována žádost o přerušení. Podprogram pro obsluhu přerušení potom může jednoduše vypočítat časový interval, pokud je znám předchozí obsah časovače T2. Druhou možností je spojit vstupy dvou střadačů paralelně, přičemž první střadač je konfigurován na reakci od vzestupné a druhý od sestupné hrany vstupního signálu. Po přerušení druhým střadačem je v prvním střadači obsah časovače T2 na začátku události a ve druhém střadači obsah T2 na konci události. Tímto způsobem byla změřena délka impulsu.

S krystalem 12 MHz má maximální doba do přetečení T2 velikost 524 ms. Pokud je měřená vnější událost kratší, počítání časových intervalů je jednoduché a přerušovací podprogram krátký. Pro delší časové intervaly musí být samozřejmě obslužná přerušovací rutina rozšířena.

Porovnávací logika

V reálném čase je obsah čítače T2 periodicky inkrementován (zvyšován o 1) a obsah každého porovnávacího registru CM0, CM1 a CM2 je porovnáván s novou hodnotou čítače T2. Pokud je nalezena shoda, na konci následujícího cyklu je nastaven příslušný odpovídající přerušovací příznak v registru TM2IR.

Pokud nastane shoda čítače T2 s CM0 a shodné bity v registru STE jsou ve stavu log. 1, řadič nastavuje bity 0 až 5 portu P4 .

Pokud nastane shoda čítače T2 s CM1 a shodné bity v registru RTE jsou ve stavu log. 1, řadič vynuluje bity 0 až 5 portu P4. Pokud jsou shodné bity v registru RTE ve stavu log. 0, potom P4.n není ovlivňován.

Pokud nastane shoda čítače T2 s CM2 a shodné bity v registru RTE jsou ve stavu log. 1, řadič překlopí bity 6 až 7 portu P4.

Dva pomocné klopné obvody TG46 a TG47 uchovávají poslední operaci. Tyto obvody mohou být čteny v registrech STE.6 a STE.7 (korespondují s P4.6 a P4.7) Bity STE.6 a STE.7 jsou určeny pouze pro čtení. Úroveň log. 1 indikuje, že následující překlopení bude nastavovat záchytný registr portu a logická 0 určuje, že záchytný registr portu bude nulován.

CM0, CM1 a CM2 jsou nulovány při hardwarovém nulování RST mikropočítače. Každý bit portu P4 může být kdykoliv nastavován nebo nulován softwarově.

Bity registru RTE mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40

kde:

TP47 - když je tento bit ve stavu log. 1, potom je P4.7 překlopen při shodě mezi CM2 a čítačem T2.

TP46 - když je tento bit ve stavu log. 1, potom je P4.6 překlopen při shodě mezi CM2 a čítačem T2.

RP45 - když je tento bit ve stavu log. 1, potom je P4.5 nulován při shodě mezi CM1 a čítačem T2.

RP44 - když je tento bit ve stavu log. 1, potom je P4.4 nulován při shodě mezi CM1 a čítačem T2.

RP43 - když je tento bit ve stavu log. 1, potom je P4.3 nulován při shodě mezi CM1 a čítačem T2.

RP42 - když je tento bit ve stavu log. 1, potom je P4.2 nulován při shodě mezi CM1 a čítačem T2.

RP41 - když je tento bit ve stavu log. 1, potom je P4.1 nulován při shodě mezi CM1 a čítačem T2.

RP40 - když je tento bit ve stavu log. 1, potom je P4.0 nulován při shodě mezi CM1 a čítačem T2.

Bity registru STE mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40

kde:

TG47 je překlápěcí bit (flip-flop).

TG46 je překlápěcí bit (flip-flop).

SP45 - když je tento bit ve stavu log. 1, potom je P4.5 nastaven při shodě mezi CM0 a čítačem T2.

SP44 - když je tento bit ve stavu log. 1, potom je P4.4 nastaven při shodě mezi CM0 a čítačem T2.

SP43 - když je tento bit ve stavu log. 1, potom je P4.3 nastaven při shodě mezi CM0 a čítačem T2.

SP42 - když je tento bit ve stavu log. 1, potom je P4.2 nastaven při shodě mezi CM0 a čítačem T2.

SP41 - když je tento bit ve stavu log. 1, potom je P4.1 nastaven při shodě mezi CM0 a čítačem T2.

SP40 - když je tento bit ve stavu log. 1, potom je P4.0 nastaven při shodě mezi CM0 a čítačem T2.

Bity registru TM2IR mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
T2OV	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0

kde:

T2OV je příznak žádosti o přerušení při šestnáctibitovém přetečení T2.

CMI2 je příznak žádosti o přerušení CM2.

CMI1 je příznak žádosti o přerušení CM1.

CMI0 je příznak žádosti o přerušení CM0.

CTI3 je příznak žádosti o přerušení CTI3.

CTI2 je příznak žádosti o přerušení CTI2.

CTI1 je příznak žádosti o přerušení CTI1.

CTI0 je příznak žádosti o přerušení CTI0.

Bity registru IP1 mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0

kde:

PT2 je úroveň přerušení přetečení T2.

PCM2 je úroveň přerušení čítače T2, komparátoru 2.

PCM1 je úroveň přerušení čítače T2, komparátoru 1.

PCM0 je úroveň přerušení čítače T2, komparátoru 0.

PCT3 je úroveň přerušení čítače T2, registru střadače 3.

PCT2 je úroveň přerušení čítače T2, registru střadače 2.

PCT1 je úroveň přerušení čítače T2, registru střadače 1.

PCT0 je úroveň přerušení čítače T2, registru střadače 0.

Časovač T3 - WATCHDOG

Na rozdíl od standardních čítačů/časovačů je v mikropočítačích 552 navíc obsažen časovač T3. Tento časovač, pokud je jeho činnost povolena, generuje systémový RESET mikropočítače, pokud program nepracuje správně.

Časovač T3 je osmibitový s jedenáctibitovou předděličkou. Předdělička je plněna kmitočtem, který odpovídá 1/12 kmitočtu oscilátoru.

Pokud hodnota osmibitového časovače přeteče, je vygenerován krátký vnitřní impuls, který je rovněž přítomen na vývodu RST mikropočítače.

WATCHDOG je aktivován, pokud je externí vývod EW (non) na nízké úrovni. Pokud je externí vývod EW (non) na nízké úrovni, je nemožné softwarově zakázat činnost WATCHDOGu.

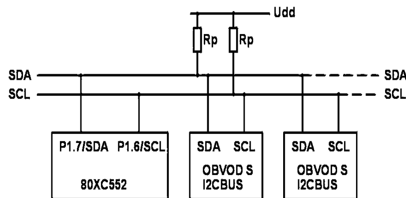
Časovač T3 musí být při své činnosti periodicky naplňován příslušnými daty. Doba mezi jednotlivými cykly plnění musí být stanovena programátorem tak, že doba přetečení T3 musí být delší než doba chodu nejdelšího úseku programu či podprogramu. Při použití krystalu 16 MHz je možné stanovit časový interval v rozmezí 1,5 až 392 ms, při krystalu 24 MHz je možné stanovit časový interval v rozmezí 1 ms až 255 ms.

Časovač WATCHDOG musí být obsluhován ve dvou krocích. Nejprve musí být nastaven bit WLE ve speciálním funkčním registru PCON, a potom teprve může být vložena nová hodnota do časovače T3. Podprogram pro obsluhu časovače T3 musí být obsluhován periodicky. Naplnění T3 hodnotou 00H dává nejdelší časový interval (510 ms pro krystal 12 MHz), naplnění T2 hodnotou 0FFH dává nejkratší interval (2 ms pro krystal 12 MHz).

Pokud je WATCHDOG v činnosti, nemůže být použit režim POWER-DOWN.

Následující příklad ukazuje práci s WATCHDOGem při středně dlouhých podprogramech:

```
                       ; DEFINICE KONSTANT
T3      EQU  0FFH      ; adresa časovače T3
PCON    EQU  087H      ; adresa SFR registru PCON
WINTV   EQU  156       ; časový interval pro Watchdog
Lcall   WATCHDOG       ; volání podprogramu
WATCHDOG: ORL  PCON,#10H  ; nastavení flagu
          MOV  T3,WINTV   ; naplnění časovače T3 hodnotou
          RET             ; návrat z podprogramu
```



Obr. 24. Typická konfigurace zapojení obvodů na sběrnici I2CBUS

Režim sníženého příkonu

Dva režimy sníženého příkonu, stejně jako u typu „51", jsou mody IDDLE a POWER-DOWN.

V modu IDDLE jsou provedeny následující funkce:

CPU je zastaven,

T2 je zastaven a vynulován,

PWM0 je vynulován a výstup je ve vysoké úrovni,

PWM1 je vynulován a výstup je ve vysoké úrovni,

ADC převod je zastaven,

T0 zůstává aktivní,

T1 zůstává aktivní,

T3 zůstává aktivní,

SIO0 zůstává aktivní,

SIO1 zůstává aktivní,

externí přerušení zůstávají aktivní.

V modu POWER-DOWN je zastaven oscilátor. Do tohoto režimu se mikropočítač uvede nastavením bitu PD v registru PCON. Režim zmrazuje všechny funkce. Pouze obsah RAM na čipu a obsah registrů SFR zůstává zachován.

Bity registru PCON mají následující význam:

D7	D6	D5	D4	D3	D2	D1	D0
SMOD	-	-	WLE	GF1	GF0	PD	IDL

kde:

SMOD nastavuje zdvojení Baudové rychlosti. Pokud je bit ve stavu log. 1, Baudová rychlost je zdvojena, když je sériový port SIO0 v modech 1, 2 nebo 3.

WLE uvolňuje přepsání obsahu T3 (WATCHDOGu). Tento příznak musí být nastaven softwarově před naplněním dat do časovače T3. Je automaticky vynulován, pokud jsou do T3 naplněna data.

GF1 je GP (General - purpose) flag bit, příznak pro rozlišení návratu z režimů s potlačeným příkonem.

GF0 je GP (General – purpose) flag bit, příznak pro rozlišení návratu z režimů s potlačeným příkonem.

PD je power down bit. Nastavením do stavu log. 1 je aktivován režim sníženého příkonu POWER -DOWN. Bit může být nastaven pouze v případě, že vstup EW (non) je ve vysoké úrovni.

IDL je bit modu IDDLE. Nastavením do stavu log. 1 je aktivován režim sníženého příkonu IDDLE.

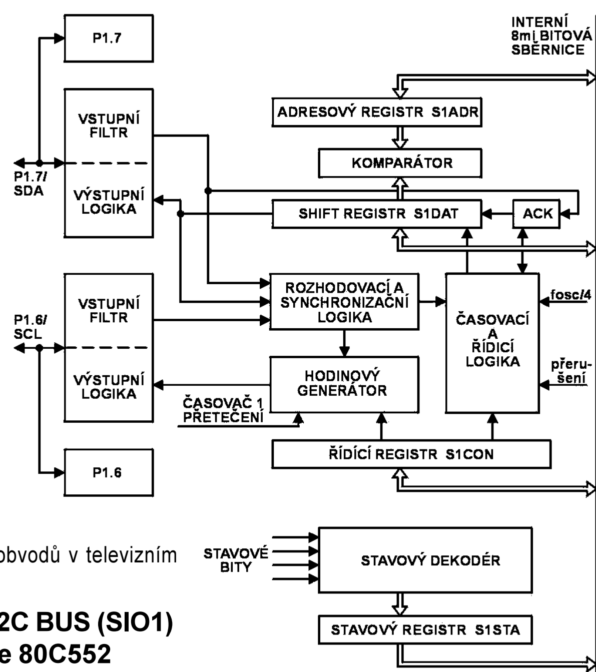
Sériové porty mikropočítače 80C552

Mikropočítače typu 8xC552 obsahují dva sériové porty SIO0 a SIO1.

SIO0 je plně duplexní sériový vstupně/výstupní port, identický se sériovým portem mikropočítače 8xC51. K tomuto portu mohou být připojeny obvody rozhraní, např. RS232 nebo RS485 pro přenos signálů na větší vzdálenosti.

SIO1 je dvoulinková sběrnice typu I2CBUS se signály SDA a SCL, sloužící k přenosu informací mezi obvody, které jsou na tuto sběrnici připojeny. Sběrnice není určena k přenosu dat na větší vzdálenosti, omezuje se pouze na obvody, umístěné na jedné desce s plošnými spoji nebo na propojení signálů mezi několika deskami v rámci jednoho zařízení. Příkladem může být propojení obvodů v televizním přijímači apod.

Obr. 25. Blokové schéma sběrnice SIO1 mikropočítače 8xC552



Sériová sběrnice I2C BUS (SIO1) mikropočítače 80C552

Hlavními rysy této sběrnice jsou:

1. Obousměrný přenos informací mezi obvody typu MASTER a SLAVE.

2. Jako MASTER nemusí být pouze jediný obvod (MULTIMASTER BUS).

3. Hodinové signály SCL synchronizují přenos mezi všemi obvody, připojenými na tuto sběrnici.

4. Signálem SDA se přenášejí data mezi obvody. Data a adresy se na této lince přenášejí v synchronizmu s hodinovým taktem na lince SCL.

5. Každá z linek SDA a SCL musí být opatřena zdvihacími (pull-up) rezistory a může být stažena do nízké úrovně každým účastníkem přenosu prostřednictvím výstupů s otevřeným kolektorem.

6. U mikropočítače 8xC552 může být tato sběrnice využita pro testování a diagnostiku.

Sběrnice I2CBUS (SIO1) mikropočítače může využívat všech přenosových modů (např. nízkorychlostních modů). Hardwarová logika SIO1 ovládá přenos byte autonomně. K tomuto účelu využívá čtyři speciální funkční registry S1CON (SIO1 řídicí registr), S1STA (SIO1 stavový registr), S1DAT (SIO1 datový registr) a S1ADR (SIO1 slave adresový registr).

Typická konfigurace zapojení obvodů na sběrnici I2CBUS je na obr. 24.

Protokol I2C má přesně definované posloupnosti, které určují start přenosu, přenos dat, potvrzení a ukončení přenosu. Podle vysílaného bitu R/W je možné rozlišit dva typy přenosů:

1. Přenos dat z vysílače MASTER do přijímače SLAVE. V klidovém stavu jsou linky SDA a SCL ve vysoké úrovni. Podmínkou zahájení přenosu je stažení linky SDA obvodem MASTER na nízkou úroveň, linka SCL zůstává na úrovni vysoké. Prvních sedm vysílaných bitů (A6 až A0) tvoří adresu, která určuje výběr obvodu SLAVE, se kterým se má komunikovat. Prvním vysílaným bitem adresy je MSB (nejvyšší bit). Následuje přenos bitu směru přenosu, v tomto případě je to log. 0. Obvod SLAVE potvrdí adresu vysláním bitu ACKNOWLEDGE a následuje přenos osmi datových bitů z obvodu MASTER do obvodu SLAVE. SLAVE potvrzuje příjem každého byte signálem ACKNOWLEDGE. MASTER může přerušit spojení podmínkou STOP, kdy SDA přejde z nízké do vysoké úrovně, přičemž SCL zůstává v úrovni vysoké.

2. Přenos dat z vysílače SLAVE do přijímače MASTER. Klidový stav linek a podmínka zahájení přenosu je stejná, jako v předchozím případě. Prvních sedm bitů adresy vysílá MASTER, následuje přenos bitu směru přenosu, v tomto případě je to log. 1. Obvod SLAVE potvrdí adresu vysláním bitu ACKNOWLEDGE a následuje přenos osmi datových bitů z obvodu SLAVE do obvodu MASTER. MASTER potvrzuje příjem každého byte signálem ACKNOWLEDGE. Konec přenosu je oznámen tak, že není vyslán bit ACKNOWLEDGE, ale je vyslána podmínka STOP (SDA přechází z nízké na vysokou úroveň, přičemž SCL zůstává na úrovni vysoké).

MASTER generuje vždy všechny hodinové impulsy včetně podmínek START a STOP. Přenos je ukončen, když je generována podmínka STOP nebo je opakována podmínka START. I když podmínka START uvolňuje přenos, pokud je opakována, nebude přenos zahájen.

V některých aplikacích může SIO1 u mikropočítače 8xC552 pracovat jako MASTER nebo jako SLAVE. Příkladem je zapojení několika mikropočítačů na jednu linku. Pokud je mikropočítač nakonfigurován jako SLAVE, jeho hardware SIO1 porovnává vysílanou adresu od obvodu MASTER se svojí přednastavenou adresou. Pokud jsou obě adresy shodné, SLAVE vykonává své interní přerušení. Pokud si mikropočítač žádá přístup na sběrnici, hardwarově čeká až do doby, kdy je sběrnice uvolněna.

Zapojení sběrnice SIO1 mikroprocesoru 88C552

Blokové schéma sběrnice SIO1 mikropočítače je na obr. 25. Celé zapojení se skládá z několika bloků, které jsou dále popsány.

Vstupní filtr a výstupní logika. Vstupní filtr je hardwarově totožný s logikou I2C. Pokud je vstupní napětí menší než 1,5 V, je stav vyhodnocován jako log. 0. Pokud je vstupní napětí větší než 3 V, je vyhodnocen stav log. 1. Vstupní signály jsou synchronizovány s interními hodinami mikropočítače ($f_{osc}/4$) a kratší impulsy na sběrnici jsou potlačeny.

Výstup tvoří otevřený kolektor tranzistoru bez vnitřní diody, připojené na napájecí sběrnici Udd. Tím je dosaženo toho, že pokud je mikropočítač připojen na sběrnici I2C a nemá připojené napájecí napětí, není činnost sběrnice I2C omezena.

Adresový registr S1ADR. Tento osmibitový registr SFR může být naplněn sedmibitovou adresou (postupně od MSB), pokud je mikropočítač konfigurován jako přijímač nebo vysílač

SLAVE. Nejnižší bit LSB (GC) je použit pro uvolnění hlavní generální volací adresy 00H.

Komparátor. Komparátor porovnává přijímanou sedmibitovou adresu s vlastní adresou, uloženou v registru S1ADR. Zároveň porovnává první přijatý byte s hlavní volací adresou 00H. Pokud je zjištěna totožnost obou adres, jsou nastaveny příslušné stavové bity a je vyvoláno přerušení.

Posuvný registr S1DAT. Tento osmibitový speciální funkční registr obsahuje data, která jsou sériově vysílána, nebo data, která byla právě přijata. Data jsou v něm posouvána zprava doleva, čili první bit, který bude vysílán, je bit MSB a při příjmu bude první přijatý bit umístěn taktéž na pozici MSB.

Řídicí a synchronizační logika. Pokud je mikroprocesor konfigurován jako MASTER, řídicí logika řídí přenos dat a tudíž součinnost výstupu dat na lince SDA s hodinovými signály na lince SCL. Zároveň je schopna konfigurovat SIO1 jako vysílač MASTER nebo přijímač SLAVE. Dovoluje také přenos dat (signály na SCL) pouze v případě, pokud přenos platného sériového byte je kompletní.

Synchronizační logika řídí činnost hodinového generátoru a synchronizuje jej s hodinovými impulsy na lince SCL, pokud hodinové signály vysílá jiné zařízení.

Hodinový generátor. Generátor je programovatelný a vystavuje hodinové řídicí impulsy na linku SCL, pokud je mikroprocesor konfigurován jako vysílač MASTER nebo přijímač MASTER. Je v nečinnosti, pokud SIO1 je ve stavu SLAVE. Výstupní hodinové impulsy mají střídu 50 %.

Časovací a řídicí logika. Tato logika přijímá signály ze synchronizační logiky, oscilátoru mikropočítače a řídicí logiky přerušení a generuje časovací a řídicí signály pro činnost posuvného registru S1DAT, uvolňuje činnost komparátoru, generuje a detekuje podmínky START a STOP, přijímá a vysílá bit ACKNOWLEDGE a řídí mody MASTER a SLAVE.

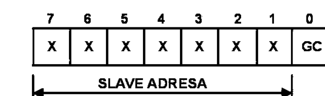
Řídicí registr S1CON. Registr je použit pro řízení portu SIO1 a konfigurují se v něm typy přenosů. Jeho funkce bude popsána dále.

Stavový dekodér a stavový registr S1STA. Stavový dekodér sbírá všechny interní stavové bity a komprimuje je na pětibitový kód. Tento kód může být např. použit pro generování vektorových adres pro rychlé procesy při obsluze žádostí o přerušení. V registru S1STA je uložen vždy kód po provedené operaci na SIO1. Kód udává stav na I2CBUS a stav hardware SIO1.

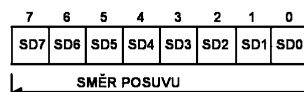
Čtyři speciální funkční registry portu SIO1. Mikropočítač 80xC552 má čtyři speciální funkční registry SFR, které se přímo podílejí na funkci rozhraní SIO1. Jsou to registry S1ADR, S1DAT, S1CON a S1STA. Pro pochopení činnosti SIO1 a vlastní programování je potřeba znát jejich funkce, a proto jsou dále popsány.

Adresový registr S1ADR. Význam jednotlivých bitů registru S1ADR je uveden na obr. 26. Registr je přímoadresovatelný na adrese 0DBH. Jeho funkce není využita, pokud mikropočítač je konfigurován jako MASTER. V modu SLAVE sedm nejvyšších bitů musí být obsazeno vlastní adresou SLAVE, a pokud je nastaven bit 0 (GC), je hlavní adresou 00H (jiná zapsaná adresa je ignorována). Stav log. 1 na příslušné pozici S1ADR vystavuje vysokou úroveň na sběrnici I2C a naopak, stav log. 0 v S1ADR vystavuje nízkou úroveň na I2C.

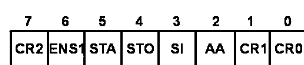
Datový registr S1DAT. Význam jednotlivých bitů registru S1DAT je uveden na obr. 27. Registr je využit pro zápis a čtení osmibitových dat. Do registru se zapisují data, která mohou být následně vyslána, nebo obsahuje právě při-



Obr. 26. Registr S1ADR (0DBH)



Obr. 27. Registr S1DAT (0DAH)



Obr. 28. Registr S1CON (0D8H)

jmutá data. Je přítomen na adrese 0DBH. Mikropočítač může svojí interní sběrnicí do tohoto registru zapisovat nebo z něho číst v případě, když SIO1 je v definovaném stavu a je nastaven příznak žádosti o přerušení SI od SIO1. Data v S1DAT jsou stabilní po celou dobu, po kterou je příznak SI nastaven. Data z registru vycházejí vždy zprava doleva, první vysílaný bit je MSB a poslední je LSB. Pokud jsou data přijímána, první přijmutý bit je umístěn na pozici MSB a postupně posouván při každém dalším přijmutém bitu k pozici LSB. S1DAT vždy obsahuje poslední byte dat, který byl přítomen na sběrnici.

Do registru může být přijato nebo z něho vysláno osm bitů dat. Stav log. 1 v S1DAT vytváří vysokou úroveň na sběrnici I2CBUS a naopak, stav log. 0 vytváří úroveň nízkou.

S1DAT a příznak ACK vytvářejí spolu devítibitový posuvný registr, který vysílá nebo přijímá osmibitová data, za kterými následuje potvrzovací bit ACKNOWLEDGE. Příznak ACK je plně řízen obvody portu SIO1 a nemůže být přístupný (hardwarově) mikropočítači. Sériová data jsou posouvána spolu s příznakem ACK náběžnými hranami hodinových impulsů na lince SCL. Pokud je přijmutý byte zapsán do S1DAT, sériová data jsou v S1DAT uvolněna a potvrzovací bit ACKNOWLEDGE je navracen řídicí logikou s devátým hodinovým impulsem. Sériová data jsou vysouvána z S1DAT sestupnou hranou hodinových impulsů na lince SCL.

Pokud se zapisují data do registru S1DAT, BSD7 je naplněn příkazem S1DAT.7, který je zároveň prvním vysílaným bitem na lince SDA.

Řídicí registr S1CON. Význam jednotlivých bitů registru S1CON je uveden na obr. 28. Registr je přístupný na adrese 0D8H. Mikropočítač může do tohoto přímo adresovatelného registru zapisovat nebo z něj může číst.

Pouze dva bity jsou ovlivňovány obvody portu SIO1. První z nich je bit SI, který se nastaví, když SIO1 žádá o přerušení. Druhý je bit STO, který se vynuluje, když je na sběrnici I2CBUS přítomna podmínka STOP. Bit STO je taktéž vynulován, když bit ENS1 je ve stavu log. 0.

Dále je uveden popis jednotlivých bitů registru S1CON:

ENS1 je uvolňovací bit sběrnice SIO1. Když je bit ENS1 ve stavu log. 0, jsou linky SDA a SCL ve stavu vysoké impedance. Vstupní signály na linkách SDA a SCL jsou ignorovány, SIO1 je v nenaadresovaném modu SLAVE a bit STO v S1CON je vynulován. V tomto stavu mohou být linky P1.6 a P1.7 využity jako běžné vstupně/výstupní linky. Linky P1.6 a P1.7 mají otevřené kolektory, a proto musí být připojené přes zdvihací rezistory k napájecí sběrnici Ucc. Když je bit ENS1 ve stavu log. 1, je uvolněno SIO1.

STA je příznak START. Když je bit STA ve stavu log. 0, nebude generována podmínka START. Když je bit STA ve stavu log. 1 v modu MASTER, SIO1 hardwarově hlídá stav sběrnice I2CBUS a generuje podmínku START, když je sběrnice uvolněna. Když uvolněna není, SIO1 čeká na podmínku STOP a po ní generuje podmínku START po prodlevě poloviny periody interního hodinového generátoru. Když je bit STA ve stavu log. 1, SIO1 je v modu MASTER a zároveň již bylo vysláno nebo přijmuto více byte, SIO1 vysílá opakovaně podmínku START. Bit STA může být nastaven vždy, a to i v případě, když SIO1 je nakonfigurováno jako SLAVE.

STO je příznak STOP. Když je bit STO ve stavu log. 0, nebude generována STOP podmínka. Když je bit STO ve stavu log. 1 a SIO1 je v modu MASTER, bude vyslána na I2CBUS podmínka STOP. Když je na sběrnici I2CBUS detekována podmínka STOP, SIO1 automaticky hardwarově vynuluje bit STO. V modu SLAVE by měl být tento bit obnoven po chybové podmínce. V tomto případě není na I2CBUS vysílána podmínka STOP. Nicméně, SIO1 se hardwarově chová tak, jako kdyby byla podmínka STOP přijata, a nastaví neadresovaný příjmový mod SLAVE. Příznak STO je nulován automaticky hardwarově.

SI je příznak žádosti o přerušení od SIO1. Když je bit SI ve stavu log. 0, není požadováno přerušení. Když je bit SI ve stavu log. 1 a současně jsou do stavu log. 1 nastaveny bity EA a ENS1, je požadováno přerušení od SIO1. Příznak SI musí být nulován softwarově.

AA je potvrzovací bit pro příznak ACKNOWLEDGE. Když je bit AA ve stavu log. 0, nebude příznak ACK navracen zpět v době impulsu ACK na SCL při přijetí dat a při SIO1 v přijímacím modu MASTER nebo v naadresovaném přijímacím modu SLAVE. Když je bit AA ve stavu log. 1, bude navracen příznak ACK v době hodinového impulsu ACK na lince SCL tehdy, když bude přijata vlastní adresa SLAVE nebo generální volací adresa (CG v S1ADR nastaven) nebo když je SIO1 v příjmovém modu MASTER a bude přijat byte dat nebo když je SIO1 v adresovaném příjmovém modu SLAVE a bude přijat byte dat.

CR0, CR1 a CR2 jsou hodinové bity, kterými se nastavuje rychlost přenosu. Rychlosti sériového přenosu (v případě, že SIO1 je v modu MASTER) jsou uvedeny v tab. 19.

Stavový registr S1STA. Tento registr je určen pouze pro čtení. Jeho poslední tři bity jsou vždy ve stavu log. 0, takže stavové kódy určuje nejvyšších pět bitů. Pokud data v S1STA nabudou hodnoty 0F8H, jsou nestabilní, SI je ve stavu log. 0 a je nutno vyčkat na platný přenos. Stav 00H informuje o chybě na sběrnici I2CBUS nebo o neplatných podmínkách START nebo STOP. Všechny ostatní hodnoty registru určují definované stavy. Při přechodu S1STA do jakéhokoliv stavu je vyvoláno přerušení od SIO1 (SI přejde do stavu log. 1) a v S1STA je přítomen platný kód.

Více informací o přenosu

Port SIO1 se může nacházet ve čtyřech operačních modech. Jsou to:

1. Vysílač MASTER.
2. Přijímač MASTER.
3. Přijímač SLAVE.
4. Vysílač SLAVE.

Vysílač MASTER. V tomto modu jsou datové byte vysílány k přijímači SLAVE.

Před přenosem musí být zapsán do registru S1CON inicializační kód v pořadí, které je uvedeno v následující tabulce:

D7	D6	D5	D4	D3	D2	D1	D0
CR2	ENS1	STA	STO	S	AA	CR1	CR0
BR2	1	0	0	0	X	BR1	BR0

kde bity na pozici BR0 až BR2 určují rychlost přenosu, bit ENS1 musí být nastaven do stavu log. 1 pro uvolnění SIO1 a na pozici AA může být zapsána log. 1 nebo log. 0 (viz předcházející popis). Bity STA, STO a SI musí být vynulovány.

Tab. 19. Nastavení přenosových rychlostí mikropočítače 8xC552

Hodnota bitů			Bitový přenos [kHz] při f_{osc}			dělicí poměr f_{osc}
CR2	CR1	CR0	6 MHz	12 MHz	16 MHz	
0	0	0	23	47	63	1/256
0	0	1	27	54	71	1/224
0	1	0	31	63	83	1/192
0	1	1	37	75	100	1/160
1	0	0	6,25	12,5	17	1/960
1	0	1	50	100	-	1/120
1	1	0	100	-	-	1/60
1	1	1	0,25 < 62,5	0,5 < 62,5	0,67 < 56	96 x (256 - hodnota čas. T1) (hodnota 0 až 254 v modu 2)

Nyní může být uvolněn přenos nastavením bitu STA instrukcí SETB. Logika SIO1 bude nyní testovat I2CBUS a generovat podmínku START, pokud bude sběrnice uvolněna. Po vyslání podmínky START je nastaven příznak (flag) SI (přerušení od SIO1) a ve stavovém registru S1STA musí být hodnota 08H. V přerušovací rutině se musí naplnit registr S1DAT adresou SLAVE a bitem, který určuje směr přenosu (SLA + W). Bit SI v registru S1CON musí být nulován před každým dalším přenosem.

Po vyslání adresy SLAVE a bitu určujícího směr přenosu a po přijetí bitu ACK je znovu nastaven příznak SI a v registru S1STA je přítomen stavový kód. Tento kód může nabývat hodnot 18H, 20H nebo 38H pro mod MASTER nebo také 68H, 78H nebo 0B0H, pokud je uvolněn mod SLAVE (bit AA je ve stavu log. 1).

Přehled stavových kódů v režimu vysílač MASTER je v tab. 20.

V tab. 20 až tab. 24 jsou použity tyto zkratky:

S je podmínka start,

SLA je sedmibitová adresa SLAVE,

R je čtený bit,

W je zapisovaný bit,

A znamená potvrzovací bit ACKNOWLEDGE,

AN znamená, že není potvrzovací bit ACKNOWL.,

DATA je osmibitový byte dat,

P je podmínka STOP,

GC je generální volací adresa (00H).

Přijímač MASTER. V tomto modu jsou byte dat přijímány z vysílače SLAVE. Před přenosem musí být zapsán do S1CON inicializační kód, který je stejný, jako u modu vysílač MASTER. Pokud byla vyslána podmínka START, přerušovací servisní rutina musí naplnit registr S1DAT sedmibitovou adresou SLAVE a bitem směru přenosu (SLA+R). Bit SI v registru S1CON musí být nulován před každým dalším přenosem. Po vyslání adresy SLAVE a bitu směru přenosu je přijmut potvrzovací bit ACK. Nyní se nastaví

Tab. 20. Stavové kódy v registru S1STA v režimu vysílač MASTER

Stav. kód	Stav I2CBUS a SIO 1	DO/Z S1DAT	DO S1CON				Akce, která následuje
			STA	STO	SI	AA	
08H	podm. START byla vyslána	naplnit SLA+W	X	0	0	X	SLA+W budou vyslány, ACK bude přijat
10H	byla opakovaně	naplnit SLA+W	X	0	0	X	SLA+W budou vyslány
	vyslána podm. START	naplnit SLA+R	X	0	0	X	SIO1 bude sepnut do modu MST/REC
18H	SLA+W byly vyslány,	naplnit byte dat	0	0	0	X	byte dat bude vyslán, ACK byl přijat
	ACK byl přijat	bez akce	1	0	0	X	opakovaná podmínka START bude vyslána
		bez akce	0	1	0	X	podmínka STOP bude vyslána, flag STO bude vynulován
		bez akce	1	1	0	X	podmínka STOP následuje po podmínce START, flag STO bude vynulován
20H	SLA+W byly vyslány,	naplnit byte dat	0	0	0	X	byte dat bude vyslán, ACK bude přijat
	ACK nebyl přijat	bez akce	1	0	0	X	opakovaná podmínka START bude vyslána
		bez akce	0	1	0	X	podmínka STOP bude vyslána, flag STO bude vynulován
		bez akce	1	1	0	X	podmínka STOP následuje po podmínce START, flag STO bude vynulován
28H	byte dat v S1DAT	naplnit byte dat	0	0	0	X	byte dat bude vyslán, ACK bude přijat
	byl vyslán, ACK byl přijat	bez akce	1	0	0	X	opakovaná podmínka START bude vyslána
		bez akce	0	1	0	X	podmínka STOP bude vyslána, flag STO bude vynulován
		bez akce	1	1	0	X	podmínka STOP následuje po podmínce START, flag STO bude vynulován
30H	byte dat v S1DAT	naplnit byte dat	0	0	0	X	byte dat bude vyslán, ACK bude přijat
	byl vyslán, ACK nebyl	bez akce	1	0	0	X	opakovaná podmínka START bude vyslána
	přijat	bez akce	0	1	0	X	podmínka STOP bude vyslána, flag STO bude vynulován
		bez akce	1	1	0	X	podmínka STOP následuje po podmínce START, flag STO bude vynulován
38H	ztráta SLA nebo	bez akce	0	0	0	X	sběrnice I2CBUS bude uvolněna
	byte dat	bez akce	1	0	0	X	podmínka START bude vyslána, pokud I2CBUS bude uvolněna

Tab. 21. Stavové kódy v registru S1STA v režimu přijímač MASTER

Stav. kód	Stav I2CBUS a SIO 1	DO/Z S1DAT	DO S1CON				Akce, která následuje
			STA	STO	SI	AA	
08H	podm. START byla vyslána	naplnit SLA+R	X	0	0	X	SLA+R budou vyslány, ACK bude přijat
10H	bude opakovaně	naplnit SLA+R	X	0	0	X	SLA+R budou vyslány
	vyslána podm. START	naplnit SLA+W	X	0	0	X	SIO1 bude sepnut do modu MST/REC
38H	ztráta bitu NOT ACK	bez akce	0	0	0	X	sběrnice I2CBUS bude uvolněna, SIO1 vstoupí do modu SLAVE
		bez akce	1	0	0	X	podmínka START bude vyslána, pokud I2CBUS bude uvolněna
40H	SLA+R byly vyslány,	bez akce	0	0	0	0	bude přijat byte dat, nebude navrácen ACK
	ACK byl přijat	bez akce	0	0	0	1	bude přijat byte dat, bude navrácen ACK
48H	SLA+R byly vyslány,	bez akce	1	0	0	X	bude vyslána opakovaně podmínka START
	ACK nebyl přijat	bez akce	0	1	0	X	bude vyslána podmínka STOP, flag STO bude vynulován
		bez akce	1	1	0	X	podmínka STOP následuje po podmínce STOP, flag STO bude vynulován
50H	byl přjat byte dat,	číst byte dat	0	0	0	0	bude přijat byte dat, nebude navrácen ACK
	ACK byl navrácen	číst byte dat	0	0	0	1	bude přijat byte dat, bude navrácen ACK
58H	byl přjat byte dat,	číst byte dat	1	0	0	X	opakovaná podmínka START bude vyslána
	ACK nebyl navrácen	číst byte dat	0	1	0	X	bude vyslána podmínka STOP, flag STO bude vynulován
		číst byte dat	1	1	0	X	podmínka STOP následuje po podmínce STOP, flag STO bude vynulován

Tab. 22. Stavové kódy v registru S1STA v režimu přijímač SLAVE

Stav. kód	Stav I2CBUS a SIO 1	DO/Z S1DAT	DO S1CON				Akce, která následuje
			STA	STO	SI	AA	
60H	vlastní SLA+W byly přijaty, ACK byl navrácen	bez akce	X	0	0	0	bude přijat byte dat, ACK nebude navrácen
		bez akce	X	0	0	1	bude přijat byte dat, ACK bude navrácen
68H	ztracená SLA jako MASTER, vlastní SLA+W byla přijata, ACK navrác.	bez akce	X	0	0	0	bude přijat byte dat, ACK nebude navrácen
		bez akce	X	0	0	1	bude přijat byte dat, ACK bude navrácen
70H	gen. adr. 00H byla přijmuta, ACK byl navrácen	bez akce	X	0	0	0	bude přijat byte dat, ACK nebude navrácen
		bez akce	X	0	0	1	bude přijat byte dat, ACK bude navrácen
78H	ztracená SLA jako MASTER, GC byla přijata, ACK byl navrácen	bez akce	X	0	0	0	bude přijat byte dat, ACK nebude navrácen
		bez akce	X	0	0	1	bude přijat byte dat, ACK bude navrácen
80H	předešlé adresování s vlastní adresou SLV, DATA byla přijata, ACK byl navrácen	čtení byte dat	X	0	0	0	bude přijat byte dat, ACK nebude navrácen
		čtení byte dat	X	0	0	1	bude přijat byte dat, ACK bude navrácen
88H	předešlé adresování s vlastní adresou SLV, DATA byla přijata, ACK nebyl navrácen	čtení byte dat	0	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC
		čtení byte dat	0	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC
		čtení byte dat	1	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC, bude vyslána podmínka START
		čtení byte dat	1	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC, bude vyslána podmínka START
90H	předešlé adresování s GC, DATA byla přijata, ACK byl navrácen	čtení byte dat	X	0	0	0	bude přijat byte dat, ACK nebude navrácen
		čtení byte dat	X	0	0	1	bude přijat byte dat, ACK bude navrácen
98H	předešlé adresování s GC, DATA byla přijata, ACK nebyl navrácen	čtení byte dat	0	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC
		čtení byte dat	0	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC
		čtení byte dat	1	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC, bude vyslána podmínka START
		čtení byte dat	1	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC, bude vyslána podmínka START
0A0H	podmínka STOP nebo opakovaná podmínka START byla přijata	bez akce	0	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC
		bez akce	0	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC
		bez akce	1	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC, bude vyslána podmínka START
		bez akce	1	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC, bude vyslána podmínka START

příznak SI a v registru S1STA je přítomen stavový kód. Tento kód může nabývat hodnot 40H, 48H nebo 38H pro mod MASTER nebo také 68H, 78H nebo 0B0H, pokud je uvolněn mod SLAVE (bit AA je ve stavu log. 1).

V tab. 21 je přehled stavových kódů v režimu přijímač MASTER.

Přijímač SLAVE. V příjmovém modu SLAVE jsou byte dat přijímány od vysílače MASTER. Pro inicializaci tohoto modu je nutné nastavit S1CON. Zároveň musí být nastavena adresa SLAVE v registru S1ADR. Horních sedm bitů registru koresponduje s adresou od vysílače MASTER a pokud je nastaven bit CG, je přednastavena generální adresa 00H a horních sedm bitů v S1ADR je ignorováno.

Nastavení registru S1CON pro inicializaci jako přijímač nebo vysílač SLAVE :

D7	D6	D5	D4	D3	D2	D1	D0
CR2	ENS1	STA	STO	SI	AA	CR1	CR0
X	1	0	0	0	1	X	X

kde bity CR0, CR1 a CR2 nemají v tomto modu význam. ENS1 musí být nastaven do stavu log. 1 pro uvolnění SIO1. Bit AA musí být nastaven pro potvrzení vlastní adresy SLAVE nebo generální adresy 00H. Bity STA, STO a SI musí být vynulovány.

Po inicializaci a nastavení adresy SIO1 přijímač čeká na shodnou adresu od vysílače MASTER. Po jejím přijmutí a zároveň při přijmutí bitu W je nastaven příznak SI a může být čten stavový kód v registru S1STA. Stejně, jako v předešlých případech, je nutné obsloužit žádost o přerušení podle platných kódů.

Pokud je bit AA vynulován během přenosu, SIO1 nenavrátí potvrzovací bit ACKNOWLED-

Tab. 23. Stavové kódy v registru S1STA v režimu vysílač SLAVE

Stav. kód	Stav I2CBUS a SIO 1	DO/Z S1DAT	DO S1CON				Akce, která následuje
			STA	STO	SI	AA	
0A8H	vlastní SLA+R byly přijaty, ACK byl navrácen	naplněn daty	X	0	0	0	bude vyslán poslední byte dat, bit ACK bude přijat
		naplněn daty	X	0	0	1	bude vyslán byte dat, bit ACK bude přijat
0B0H	ztracená SLA jako MASTER, vlastní SLA+W byla přijata, ACK navrác.	naplněn daty	X	0	0	0	bude vyslán poslední byte dat, bit ACK bude přijat
		naplněn daty	X	0	0	1	bude vyslán byte dat, bit ACK bude přijat
0B8H	byte dat v reg. S1DAT byl vyslán, ACK byl přijat	naplněn daty	X	0	0	0	bude vyslán poslední byte dat, bit ACK bude přijat
		naplněn daty	X	0	0	1	bude vyslán byte dat, bit ACK bude přijat
0C0H	byte dat v S1DAT byl vyslán, ACK nebyl přijat	bez akce	0	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC
		bez akce	0	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC
		bez akce	1	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC, bude vyslána podmínka START
		bez akce	1	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC, bude vyslána podmínka START
0C8H	poslední byte dat v reg. S1DAT byl vyslán (AA = 0), ACK byl přijat přijat	bez akce	0	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC
		bez akce	0	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC
		bez akce	1	0	0	0	sepnut neadresovaný mod SLV, neuznána adr. SLA nebo GC, bude vyslána podmínka START
		bez akce	1	0	0	1	sepnut neadresovaný mod SLV, uznána adr. SLA nebo GC bude vyslána podmínka START

Tab. 24. Stavové kódy v registru S1STA při nedefinovaných stavech na sběrnici I2CBUS

Stav. kód	Stav I2CBUS a SIO 1	DO/Z S1DAT	DO S1CON				Akce, která následuje
			STA	STO	SI	AA	
0F8H	neplatné stavy, SI = 0	bez akce	bez	akce			čeká nebo proces platného přenosu
00H	chyba na sběrnici nebo neplatné podmínky START nebo STOP, SIO1 vstoupilo do nedefin. stavu	bez akce	0	1	0	X	sběrnice je uvolněna a SIO1 je sepnuto do neadresovaného modu SLV, STO je vynulován

GE na lince SDA. Pokud je AA vynulován, SIO1 nekoresponduje s vlastní adresou SLAVE ani s generální volací adresou, avšak sběrnice I2CBUS je stále monitorována. Po nastavení bitu AA pracuje zařízení opět ve svém modu. Takže ovládáním bitu AA může být na určitou dobu izolováno SIO1 od I2CBUS.

V tab. 22 je přehled stavových kódů v režimu přijímač SLAVE.

Vysílač SLAVE. V tomto modu jsou byte dat vysílány k přijímači MASTER. Přenos dat je inicializován shodným způsobem jako v přijímači SLAVE. Když jsou registry S1ADR a S1CON inicializovány, čeká SIO1 tak dlouho, dokud není přijata vlastní adresa SLAVE a po ní následující bit R, který udává směr přenosu (tento bit musí být v úrovni log. 1). Po přijetí vlastní adresy SLAVE a bitu R je nastaven příznak SI a v registru S1STA je platný stavový kód. Stejně, jako v předešlých případech, je nutné obsloužit žádost o přerušení.

Pokud bude bit AA vynulován během přenosu, SIO1 vyšle poslední byte a v registru S1STA bude kód 0C0H nebo 0C8H. SIO1 je sepnuto do neadresovaného modu SLAVE a bude ignorovat přijímač MASTER. Pokud je bit AA vynulován, SIO1 nekoresponduje s vlastní adresou SLAVE ani s generální volací adresou, ale sběrnice I2CBUS je stále monitorována. Po nastavení AA pracuje zařízení opět ve svém modu. Z toho vyplývá, že ovládáním bitu AA může být na určitou dobu izolováno SIO1 od sběrnice I2CBUS.

V tab. 23 je přehled stavových kódů v režimu vysílač SLAVE.

Nedefinované stavy. V nedefinovaných stavech může registr S1STA nabýt hodnot 0F8H nebo 00H. V obou stavech je vyhlášena chyba na sběrnici I2CBUS. Popis těchto stavů je v tab. 24.

Výstupy s šířkově modulovanými impulsy

Mikropočítač 552 obsahuje dva výstupy s šířkově modulovanými impulsy. Na výstupech jsou impulsy s programovatelnou délkou a střídou.

Opakovací kmitočet je definován v osmibitové předděličce PWMP, která ovládá hodinový vstup čítače. Osmibitový čítač čítá v modulu 255 (od 0 do 254). Hodnota osmibitového čítače je porovnávána s obsahem dvou registrů PWM0 a PWM1. Pokud obsah každého z těchto registrů je větší než hodnota čítače, odpovídající výstup PWM0 (non) nebo PWM1 (non) je nastaven na nízkou úroveň. Jestliže obsah některého z těchto registrů je stejný nebo nižší, je odpovídající výstup nastaven na úroveň vysokou. Střída impulsů je tedy definována registry PWM0 a PWM1. Střída se může měnit v rozmezí od 0 do 1 a může být programována po krocích 1/255. Bity 0 až 7 určují dělicí faktor, který je PWMP + 1.

Bity 0 až 7 registrů PWM0 a PWM1 určují střídu. Poměr nízké ku vysoké úrovni z registru PWMn odpovídá poměru PWMn / (255 – PWMn).

Výstupy PWM mohou být použity pro řízení stejnosměrných motorů. Rychlost otáčení motorů může být potom řízena plynule a je dána obsahem registrů PWMn.

Výstupy PWM se mohou využít i jako dva D/A převodníky.

Opakovací kmitočet je určen vztahem:

$f_{PWM} = f_{osc} / [2 \times (1 + PWMP) \times 255$.

Ze vztahu vyplývá, že opakovací kmitočet může být v rozsahu od 123 Hz do 31,4 kHz při použití krystalu 16 MHz nebo v rozsahu od 184 Hz do 47,1 kHz při použití krystalu 24MHz.

Při hodnotách 00H nebo 0FFH v registrech PWM zůstávají příslušné výstupní kanály v úrovních log. 1 nebo log. 0.

Struktura výstupních kanálů s šířkově modulovanými impulsy je na obr. 29.

Analogová vstupní část

Mikropočítač 80C552 má osm multiplexovaných analogových vstupů s desetibitovým A/D převodníkem.

A/D převodník pracuje na principu postupné aproximace. Převodník obsahuje D/A převodník, který převádí obsah aproximačního registru na napětí, které je porovnáváno se vstupním analogovým napětím *Vin*. Výstupem komparátoru se nazpět řídí aproximační registr.

Funkce analogové vstupní části vyplývá z popisu bitů registru ADCON, který ji řídí.

Bity registru ADCON mají nasledující význam:

D7	D6	D5	D4	D3	D2	D1	D0
ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0

kde:

ADC.1 a ADC.0 jsou bity 1 a 0 výsledku A/D převodu.

ADEX je bit, který uvolňuje start převodu vnějším signálem, který je přiváděn na vývod STADC mikropočítače. Je-li bit ADEX ve stavu log. 0, spouští se převod pouze softwarově (nastavením bitu ADCS), je-li bit ADEX ve stavu log. 1, spouští se převod softwarově nebo externě nástupnou hranou signálu na vývodu STADC.

ADCI je příznak žádosti o přerušení, který je nastaven po převodu, kdy je možné číst hodnotu po převodu. Příznak by měl být nulován v podprogramu pro obsluhu přerušení. Pokud je tento příznak nastaven, převodník AD nemůže spustit nový převod. Bit ADCI nemůže být nastaven do stavu log. 1 softwarově.

ADCS je bit, jehož nastavením do stavu log. 1 se spouští (startuje) převod. Bit může být nastaven softwarově nebo externě z vývodu STADC. Interní logika zabezpečuje, že tento bit je na vysoké úrovni po celou dobu převodu. Po ukončení převodu je bit ADCS nulován. Bit nemůže být nulován softwarově. Pokud jsou bity ADCS a ADCI ve stavu log. 1, nemůže být spuštěn nový převod.

AADR2, AADR1 a AADR0 jsou bity pro výběr jednoho z osmi analogových vstupů (viz tab. 25). Výběr může být změněn pouze v případě, když jsou bity ADCI a ADCS ve stavu log. 0.

Stavy A/D převodníku v závislosti na stavu bitů ADCI a ADCS jsou shrnuty v tab. 26.

Úplný převod trvá 50 strojových cyklů (u typu 8x562 trvá převod 24 strojových cyklů).

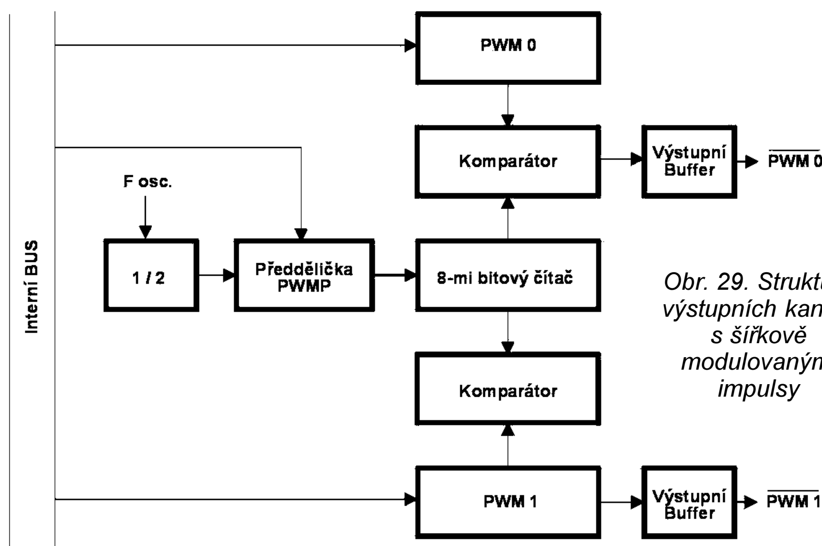
Převodník má svůj vlastní vstup napájecího napětí a dva vstupy pro napětí referenční. Referenční napětí se musí pohybovat v intervalu napájecího napětí převodníku a může jej překročit v kladném i v záporném smyslu o 0,2 V.

Tab. 26. Stavy A/D převodníku

ADCI	ADCS	ADC STAV
0	0	ADC připraven, může být nastartován nový převod
0	1	ADC není připraven, nemůže být nastartován nový převod
1	0	převod je kompletní, nový převod je možný po vynulování bitu ADCI
1	1	převod je kompletní, nový převod je možný po vynulování bitu ADCI

Tab. 25. Výběr analogového vstupu

AADR2	AADR1	AADR0	Vybraný kanál
0	0	0	ADC0 (P5.0)
0	0	1	ADC1 (P5.1)
0	1	0	ADC2 (P5.2)
0	1	1	ADC3 (P5.3)
1	0	0	ADC4 (P5.4)
1	0	1	ADC5 (P5.5)
1	1	0	ADC6 (P5.6)
1	1	1	ADC7 (P5.7)



Obr. 29. Struktura výstupních kanálů s šířkově modulovanými impulsy

Instrukční soubor mikropočítačů řady „51”

V této kapitole je popsán instrukční soubor mikropočítačů řady „51” a odvozených typů. Je nutné si uvědomit, že první bit osmibitové konstanty je označován pořadovou číslicí 0 a poslední osmý bit pořadovou číslicí 7 (např. ACC.0 až ACC.7). Totéž se týká např. registrů dané banky, kdy první registr je označován jako R0 a poslední R7.

Vzhledem k adresovacím schopnostem mikropočítačů řady „51” jsou rozlišeny tyto varianty:

Rn - Registrové adresování, kde číslo *n* (0 až 7) udává registr, použitý pro danou operaci.

@Ri - Nepřímé adresování, kde číslo *i* udává registr (0 až 1), který bude použit jako osmibitová adresa.

@DPTR - Nepřímé adresování, kde tento zápis se používá pro šesnáctibitové adresování programu nebo externí paměti dat.

Direct - Přímé adresování, kde tento zápis přímo označuje byte ze spodní poloviny vnitřní RAM mikropočítače nebo některý SFR (speciální funkční registr). Direct adresa vnitřní RAM nebo SFR se zapisuje jako další byte za operačním kódem instrukce a je osmibitová.

Bit - Zápis Bit označuje bit z bitově adresovatelné oblasti vnitřní RAM nebo z bitově adresovatelného SFR. Přímá adresa bitu je osmibitová a zapisuje se jako další byte za operačním kódem instrukce.

#data - Označení přímých osmibitových dat, která vstupují do operace jako jeden z operandů. Zapisují se jako jeden nebo dva byte za operačním kódem instrukce.

Zápis konstant může být ve formě dekadického, hexadecimálního nebo binárního tvaru. Ve vyjímečných případech některých překladačů i ve formě tvaru oktalového. Osmibitová hodnota představuje rozmezí čísel od 0 do 255, šesnáctibitová hodnota představuje rozmezí čísel od 0 do 65535 v dekadickém tvaru.

Dekadický tvar se vyjadřuje hodnotou s koncovým písmenem D, např 255D.

Hexadecimální tvar se vyjadřuje hodnotou s koncovým písmenem H, např. 55H. Pokud je první číslice zastoupena písmenem, je nutné jako první číslici uvést nulu. Např. nelze zapsat E3H ale je nutno zapsat 0E3H.

Binární tvar vyjadřuje hodnotou s koncovým písmenem B, např. 01000110B.

Příklad převodu mezi jednotlivými soustavami:

0D = 0H = 0B a lze ve všech případech většinou zapsat pouze 0;

255D = 0FFH = 11111111B;

100D = 64H = 01100100B.

Typy adresování

Přímé adresování registry. Je základním adresováním. Při tomto způsobu se poslední tři bity operačního kódu využívají jako adresa jednoho z registrů R0 až R7 vybrané banky. Kromě toho lze adresovat též akumulátorem Acc, registrem B, DPTR a CY. Přímé adresování lze použít pro přístup k dolním 128 byte vnitřní paměti RAM a ke všem speciálním funkčním registrům SFR. Operand instrukce obsahuje adresu v rozmezí 00H až 0FFH.

Nepřímé adresování registry. Je určeno pro přístup do vnitřní nebo vnější paměti dat v rozmezí adres 0000H až 00FFH. Pro toto adresování lze vyuřít registrů R0 a R1 vybrané banky. Pokud je adresována nepřímo vnitřní paměť RAM, je dostupných pouze 128 byte u řady „51” a celých 256 byte u řady „32” či „552”. Změnou operačního kódu lze u všech mikropočítačů adresovat 256 byte vnější paměti dat. Registry SFR nejsou nepřímým adresováním přístupny. Přístup do dalších stránek vnější paměti dat lze uskutečnit šestnáctibitovým registrovým párem DPTR (DPL a HPH). Do skupiny nepřímého adresování patří též instrukce PUSH a POP, které ukládají nebo naplňují obsah adresovaného registru do nebo ze zásobníku. Ukazatel zásobníku může ukazovat kamkoliv do vnitřní RAM.

Bezprostřední adresování. Toto adresování umožňuje plnit registry nebo paměťová místa obsahem operandu, který následuje za operačním kódem. Tímto způsobem lze definovat osmibitové a šestnáctibitovébitové konstanty programu.

Rozšířené adresování. Toto adresování používá jako adresu druhý nebo třetí byte instrukce (dlouhý skok LJMP) nebo je horní část rozšířené adresy uložena do dolních tří bitů operačního kódu a nižších osm bitů adresy je obsaženo v následujícím operandu za operačním kódem (krátký skok LJMP).

Relativní adresování. Používá se prakticky u všech pomíněných skoků. Operand, následující za operačním kódem nebo druhým byte instrukce, má význam posunu adresy o hodnotu sedmibitového binárního čísla se znaménkem. Lze provést skok o 127 adres zpět nebo o 128 adres vpřed od instrukce pomíněného skoku.

Nepřímé adresování s bázovým registrem. Toto adresování umožňuje jednak adresovat data v paměti programu relativně vzhledem k bázovému registrovému páru DPTR nebo adresovat relativně data v paměti programu vzhledem k programovému čítači PC jako bázovému registru. Adresový byte z paměti programu je uložen do akumulátoru místo osmibitového offsetu (posunu) adresy. Tento způsob adresování umožňuje vybírat data z tabulek, umístěných v paměti programu.

Bitové adresování. Pomocí bitového adresování lze ovlivnit příznak přetečení CY a rovněž lze ovlivnit jednotlivé bity v paměti RAM, kde je vymezen blok adres od 20H do 2FH. Tyto jednotlivé bity lze nastavovat do stavu log. 1 nebo je nulovat. Kromě těchto 128 bitů lze bitově ovlivnit obsah většiny speciálních funkčních registrů SFR. Do skupiny instrukcí bitového adresování patří i instrukce podmíněných skoků, které větví program podle hodnoty adresovaného bitu nebo pokud je příslušný bit nastaven, provedou odskok v programu a tento bit vynulují.

Význam použitých symbolů

A, ACC	Akumulátor (střadač).
AC	Auxiliary Carry - pomocný přenos. Nastaví se při přenosu z 3. do 4. bitu.
C	Carry, bit přetečení. Nastaví se při přenosu ze 7. bitu.
DPTR	Šestnáctibitový pár registrů DPL a¨DPH.
DPL	Spodní část DPTR (bity 0 až 7).
DPH	Horní část DPTR (bity 8 až 15).
OV	Indikuje kladný výsledek ze dvou záporných vstupů nebo záporný výsledek ze dvou kladných vstupů. OV = 1, pokud byl přenos z 6. do 7. bitu, ale nebyl přenos ze 7. bitu, OV = 1, pokud byl přenos ze 7. bitu, ale nebyl přenos z 6. do 7. bitu, OV = 0, ve všech ostetních případech. Při odečítání SUBB mají všechny přenosy opačný význam.
PC	Programový čítač.
PSW	Registr stavového slova.
Rn	Pracovní registr (R0 až R7) určené banky registrů.
@Ri	Registr, který svou hodnotou adresuje pametové místo. K tomuto adresování lze využít pouze registry R0 a R1.
SFR	Speciální funkční registr.
SP	Ukazatel zásobníku.
#data	Osmibitová přímá konstanta (přímá data).

Aritmetické operace

Instrukce pracují s ALU (aritmeticko – logická jednotka) a provádějí jednoduché matematické operace. Při svojí činnosti mění některé příznaky - bity registru PSW (stavové slovo). Jako jeden z operandů využívají vždy akumulátor ACC. Po ukončení operace je obsah akumulátoru přepsán výsledkem a jeho původní hodnota se ztrácí.

ADD A,#data	**Přičtení přímých dat k akumulátoru A**

Funkce	ACC = ACC + data
Popis	Je sečtena hodnota akumulátoru s osmibitovou přímou konstantou a výsledek je uložen do akumulátoru
Počet cyklů	1
Počet byte	2
Strojový kód	024H
Ovlivněné příznaky	C, AC, OV, P
1. byte	ADD A,#data
2. byte	data

Příklad: ADD A,#07 ; přičtení hodnoty 07H k akumulátoru

ADD A,@Ri	**Přičtení obsahu dat v paměti k akumulátoru A**

Funkce	ACC = ACC + (Ri)
Popis	Obsah rezidentní paměti dat naadresované registrem Ri je přičten do akumulátoru A
Počet cyklů	1
Počet byte	1
Strojový kód	0010011i (strojový kód = = 026H až 027H podle použitého registru R0 až R1)
Ovlivněné příznaky	C, AC, OV, P
Použitelné registry	R0, R1
1. byte	ADD A,@Ri

```
Příklad: MOV R0,#2FH  ; přesun zvolené
                        adresy do R0
         ADD A,@R0    ; sečtení obsahu dat
                        na adrese 2FH
                        s akumulátorem
```

ADD A, Rn	**Přičtení obsahu registru k akumulátoru A**

Funkce	ACC = ACC + Rn
Popis	Obsah registru Rn je přičten k akumulátoru A
Počet cyklů	1
Počet byte	1
Strojový kód	00101nnn (strojový kód = = 028H až 02FH podle použitého registru R0 až R7)
Ovlivněné příznaky	C, AC, OV, P
Použitelné registry	R0, R1, R2, R3, R4, R5, R6, R7
1. byte	ADD A,Rn

Příklad: ADD A,R2 ; sečtení obsahu registru R2 s akumulátorem

ADD A,direct	**Přičtení obsahu dat ve spodní polovině vnitřní RAM nebo některým ze SFR s akumulátorem A**

Funkce	ACC = ACC + direct
Popis	Přímým adresováním se obsah bytu ze spodní poloviny vnitřní RAM nebo obsah některého speciálního funkčního registru SFR přičte k akumulátoru A
Počet cyklů	1
Počet byte	2
Strojový kód	025H
Ovlivněné příznaky	C, AC, OV, P
1. byte	ADD A,dirrect
2. byte	dirrect adr.

ADDC A,#data	**Přičtení přímých dat a bitu přenosu k akumulátoru A**

Funkce ACC = ACC + data + C
Popis Data, určená přímým operandem jsou přičtena k akumulátoru A zároveň s bitem přenosu C
Počet cyklů 1
Počet byte 2
Strojový kód 034H
Ovlivněné příznaky C, AC, OV, P
1. byte ADDC A,#data
2. byte data

Příklad: ADDC A,#0F7H ; součet bitu přenosu C a přímé hodnoty F7H s akumulátorem

ADDC A,@Ri	**Přičtení obsahu dat v paměti a bitu přenosu C k akumulátoru A**

Funkce ACC = ACC + (Ri) + C
Popis Obsah rezidentní paměti dat naadresované registrem Ri je přičten k akumulátoru A spolu s bitem přenosu C
Počet cyklů 1
Počet byte 1
Strojový kód 0011011i (strojový kód = =036H až 037H podle použitého registru R0 až R1)
Ovlivněné příznaky C, AC, OV, P
Použitelné registry R0, R1
1. byte ADDC A,@Ri

Příklad: MOV R1,#40D ; přesun adresy do registru R1
ADDC A,@R1 ; součet bitu C a obsahu adresy 40D (dekadicky) s akumulátorem

ADDC A,Rn	**Přičtení obsahu registru a bitu přenosu C k akumulátoru A**

Funkce ACC = ACC + Rn + C
Popis Obsah registru Rn je přičten k akumulátoru A spolu s bitem přenosu C
Počet cyklů 1
Počet byte 1
Strojový kód 00111nnn (strojový kód = = 038H až 03FH podle použitého registru R0 až R7)
Ovlivněné příznaky C, AC, OV, P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7
1. byte ADDC A,Rn

Příklad: ADDC A,R7 ; součet bitu C a obsahu registru R7 s akumulátorem

ADDC A,direct	**Přičtení obsahu dat ve spodní polovině vnitřní RAM nebo některým ze SFR s akumulátorem A a zároveň s bitem přenosu C**

Funkce ACC = ACC + direct + C
Popis Přímým adresováním se obsah bytu ze spodní poloviny vnitřní RAM nebo obsah některého speciálního funkčního registru SFR přičte k akumulátoru spolu s hodnotou bitu přenosu C
Počet cyklů 1
Počet byte 2
Strojový kód 035H
Ovlivněné příznaky C, AC, OV, P
1. byte ADDC A,dirrect
2. byte dirrect adr.

DA A	**Dekadická oprava akumulátoru A**

Funkce if (ACC (0..3) > 9) or (AC = 1) then ACC = ACC + 6
if (ACC (4..7) > 9) or (C = 1) then ACC = ACC + 60H, C = 1
Popis Osmibitová hodnota v akumulátoru A je převedena na dvě čtyřbitová BCD čísla. Bit C se při této operaci nastavuje podle jejího výsledku. Je-li obsah bitů 0 až 3 větší než 9, nebo je-li AC roven 1, je k obsahu akumulátoru připočteno 6. Pak jsou testovány bity 4 až 7. Je-li jejich obsah větší než 9, nebo je-li C nastaven na 1, tyto bity jsou zvětšeny o 6. Proběhne-li přitom přenos (přetečení), C je nastaven na 1, jinak je nulován. Funkce se musí používat bezprostředně po instrukci ADD, protože využívá ke své funkci příznaky C a AC
Počet cyklů 1
Počet byte 1
Strojový kód 0D4H
Ovlivněné příznaky C, AC
1. byte DA A

DIV AB	**Celočíselné dělení**

Funkce PC = PC + 1
if B = 0 then OV = 1 else OV = 0
ACC = ACC div B (ACC = výsledek)
B = ACC mod B (B = zbytek po dělení)
C = 0
Popis Je provedeno celočíselné dělení akumulátoru A registrem B. Výsledek je uložen do ACC, zbytek po dělení do registru B. Příznak OV se nastavuje při dělení nulou, C je vždy nulový
Počet cyklů 4
Počet byte 1
Strojový kód 084H
Ovlivněné příznaky C, P, OV
1. byte DIV AB

Příklad: MOV A,#100D ; naplnění akumulátoru dekadickou hodnotou 100
MOV B,#20D ; naplnění registru B dekadickou hodnotou 20
DIV AB ; dělení akumulátoru A registrem B (výsledek 5 v akumulátoru)

MUL AB	**Násobení**

Funkce PC = PC + 1
ACC = ACC x B (bity 0 až 7)
B = ACC x B (bity 8 až 15)
C = 0
Popis Je provedeno vzájemné násobení akumulátoru A s registrem B. Šesnáctibitový výsledek je uložen zpět do akumulátoru A a registru B. Příznak C je vždy nulován, příznak OV je nastaven pokud je výsledek větší, než 0FFH
Počet cyklů 4
Počet byte 1
Strojový kód 0A4H
Ovlivněné příznaky C, P, OV
1. byte MUL AB

Příklad: MOV A,#10D ; naplnění akumulátoru dekadickou hodnotou 10
MOV B,#20D ; naplnění registru B dekadickou hodnotou 20
MUL AB ; násobení akumulátoru A registrem B

SUBB A,Rn	**Odečtení obsahu registru a bitu přenosu C od hodnoty akumulátoru A**

Funkce PC = PC + 1
ACC = ACC - Rn - C
Popis Instrukce odečte od hodnoty v A obsah regisrtu Rn a zároveň obsah C
Počet cyklů 1
Počet byte 1
Strojový kód 10011nnn (strojový kód = = 098H až 09FH dle použitého registru R0 až R7)
Ovlivněné příznaky C, AC, OV, P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7
1. byte SUBB A,Rn

Příklad: MOV A,#100D ; naplnění akumulátoru dekadickou hodnotou 100
MOV R6,#20D ; naplnění registru R6 dekadickou hodnotou 20
SUBB A,R6 ; odčítání hodnoty v registru R6 od akumulátoru

SUBB A,direct	**Odečtení dirrect a bitu přenosu od hodnoty akumulátoru A**

Funkce PC = PC + 2
ACC = ACC - direct - C
Popis Instrukce odečte od hodnoty v A hodnotu direct a obsah C
Počet cyklů 1
Počet byte 2
Strojový kód 095H
Ovlivněné příznaky C, AC, OV, P
1. byte SUBB A,dirrect
2. byte dirrect adr.

SUBB A,@Ri	**Odečtení obsahu paměti naadresované registrem Ri a bitu přenosu od akumulátoru**

Funkce PC = PC + 1
ACC = ACC – (Ri) - C
Popis Obsah rezidentní paměti dat naadresované registrem Ri je odečten do akumulátoru A spolu s bitem přenosu C
Počet cyklů 1
Počet byte 1
Strojový kód 1001011i (strojový kód = = 096H až 097H podle použitého registru R0 až R1)
Ovlivněné příznaky C, AC, OV, P
Použitelné registry R0, R1
1. byte SUBB A,@Ri

SUBB A,#data	**Odečtení přímých dat a bitu přenosu od akumulátoru A**

Funkce PC = PC + 2
ACC = ACC – data - C
Popis Instrukce odečte přímá data od akumulátoru A spolu s bitem přenosu C
Počet cyklů 1
Počet byte 2
Strojový kód 094H
Ovlivněné příznaky C, AC, OV, P
1. byte SUBB A,#data
2. byte data

Příklad: MOV A,#100D ; naplnění akumulátoru dekadickou hodnotou 100
SUBB A,#10D ; odčítání přímé hodnoty 10D od akumulátoru

Logické instrukce

Tyto instrukce provádějí logické operace s osmibitovými operandy. Nemají vliv na příznakové bity. Jsou zastoupeny logickými funkcemi AND, OR a XOR (exclusive OR). Kromě vlastní funkce lze logické funkce s výhodou využít pro tzv. maskování, kdy je zapotřebí měnit pouze některé bity slova a neovlivnit bity ostatní.

ANL – logický součin AND

ORL – logický součet OR

XLR – logický EXCLUSIVE OR

CLR – nulování akumulátoru
CPL – inverze akumulátoru

ANL A,#data — Logický součin akumulátoru A s přímou hodnotou

Funkce PC = PC + 2
ACC = ACC and DATA
Popis Mezi obsahem akumulátoru A a maskou, přímo určenou operandem se provede logický součin AND. Výsledek je zapsán do akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 054H
Ovlivněné příznaky P

1. byte ANL A,#data
2. byte data

Příklad: ANL A,#00011010B ; funkce AND s maskou 0001101B (dvojk. soust.)

ANL A,@Ri — Logický součin akumulátoru A a pamětí dat

Funkce PC = PC + 1
ACC = ACC and (Ri)
Popis Mezi obsahem akumulátoru A a obsahem paměti dat je provedena logická funkce AND. Výsledek je uložen do akumulátoru A, obsah adresy paměti dat se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 0101011i (strojový kód = = 056H až 057H podle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1

1. byte ANL A,@Ri

Příklad: MOV R0,#0EFH ; přesun přímé hodnoty EFH do R0
ANL A,@R0 ; funkce AND s obsahem akumulátoru a maskou na adrese EFH

ANL A,Rn — Logický součin akumulátoru A a registru

Funkce ACC = ACC and Rn
Popis Mezi akumulátorem A a pracovním registrem Rn (maskou) je provedena logická funkce AND. Výsledek je uložen do akumulátoru A, obsah pracovního registru se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 01011nnn (strojový kód = = 058H až 05FH podle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte ANL A,Rn

Příklad: ANL A,R5 ; AND obsahu akumulátoru a maskou v registru R5

ANL A,direct — Logický součin akumulátoru A a direct

Funkce PC = PC + 2
ACC = ACC and direct
Popis Instrukce provede logický součin akumulátoru A a hodnoty direct
Počet cyklů 1
Počet byte 2
Strojový kód 055H
Ovlivněné příznaky P

1. byte ANL A,direct
2. byte přímá adresa do paměti nebo SFR

ANL direct,#data — Logický součin direct a přímých dat

Funkce PC = PC + 3
Direct = direct and data
Popis Instrukce provede logický součin hodnoty direct a přímých dat
Počet cyklů 2
Počet byte 3
Strojový kód 053H
Ovlivněné příznaky žádné

1. byte ANL direct,#data
2. byte přímá adresa do paměti nebo SFR
3. byte data

ANL direct,A — Logický součin direct a akumulátoru A

Funkce PC = PC + 2
Direct = direct and ACC
Popis Instrukce provede logický součin direct a akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 052H
Ovlivněné příznaky žádné

1. byte ANL direct,A
2. byte direct adr.

CLR A — Nulování akumulátoru A

Funkce PC = PC + 1
ACC = 00H
Popis Po vykonání této instrukce je obsah akumulátoru A nulový
Počet cyklů 1
Počet byte 1
Strojový kód 0E4H
Ovlivněné příznaky P

1. byte CLR A

CPL A — Doplněk střadače

Funkce ACC = not ACC
Popis Obsah akumulátoru je komplementován, je vytvořen doplněk. Jedná se o první doplněk, kdy každá jednička je nahrazena nulou a každá nula jedničkou
Počet cyklů 1
Počet byte 1
Strojový kód 0F4H
Ovlivněné příznaky P

1. byte CPL A

Příklad: MOV A,#10101010B ; nalnění akumulátoru bitovou hodnotou 10101010B
CPL A ; nyní je v akukulátoru hodnota 01010101B

ORL A,#data — Logický součet akumulátoru A s přímou hodnotou

Funkce PC = PC + 2
ACC = ACC or DATA
Popis Mezi obsahem akumulátoru A a maskou, přímo určenou operandem se provede logický součet OR. Výsledek je zapsán do akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 044H
Ovlivněné příznaky P

1. byte ORL A,#data
2. byte data

Příklad: ORL A,#00011010B ; funkce OR s maskou 0001101B (dvojková soustava)

ORL A,@Ri — Logický součet akumulátoru A a pamětí dat

Funkce PC = PC + 1
ACC = ACC or (Ri)
Popis Mezi obsahem akumulátoru A a obsahem paměti dat je provedena logická funkce OR. Výsledek je uložen do akumulátoru A, obsah adresy paměti dat se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 0100011i (strojový kód = = 046H až 047H podle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1

1. byte ORL A,@Ri

Příklad: MOV R0,#0EFH ; přesun přímé hodnoty EFH do R0
ORL A,@R0 ; funkce OR s obsahem akumulátoru a maskou na adrese EFH

ORL A,Rn — Logický součet akumulátoru A a registru

Funkce PC = PC + 1
ACC = ACC or Rn
Popis Mezi akumulátorem A a pracovním registrem Rn (maskou) je provedena logická funkce OR. Výsledek je uložen do akumulátoru A, obsah pracovního registru se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 01001nnn (strojový kód = = 048H až 04FH podle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte ORL A,Rn

Příklad: ORL A,R5 ; OR obsahu akumulátoru a maskou v registru R5

ORL A,direct — Logický součet akumulátoru A a direct

Funkce PC = PC + 2
ACC = ACC or Direct
Popis Instrukce provede logický součet akumulátoru A a hodnoty Direct
Počet cyklů 1
Počet byte 2
Strojový kód 045H
Ovlivněné příznaky P

1. byte ORL A,Direct
2. byte direct adr.

ORL direct,#data — Logický součet Direct a přímých dat

Funkce PC = PC + 3
Direct = Direct or Data
Popis Instrukce provede logický součet hodnoty Direct a přímých dat
Počet cyklů 1
Počet byte 3
Strojový kód 043H
Ovlivněné příznaky žádné

1. byte ORL direct,#data
2. byte přímá adresa do paměti nebo SFR
3. byte data

ORL direct,A — Logický součet direct a akumulátoru A

Funkce PC = PC + 2
Direct = Direct or ACC
Popis Instrukce provede logický součet Direct a akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 042H
Ovlivněné příznaky žádné

1. byte ORL direct,A
2. byte direct adr.

XRL A,#data — Nonekvivalence akumulátoru A s přímou hodnotou

Funkce PC = PC + 2
ACC = ACC xor DATA

Popis Mezi obsahem akumulátoru A a maskou, přímo určenou operandem se provede logická funkce XOR. Výsledek je zapsán do akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 064H
Ovlivněné příznaky P

1. byte XRL A,#data
2. byte data

XRL A,@Ri	Nonekvivalence akumulátoru A a pamětí dat

Funkce PC = PC + 1
ACC = ACC xor (Ri)
Popis Mezi obsahem akumulátoru A a obsahem paměti dat je provedena logická funkce XOR. Výsledek je uložen do akumulátoru A, obsah adresy paměti dat se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 0110011i (strojový kód = = 066H až 067H podle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1

1. byte XRL A,@Ri

XRL A,Rn	Nonekvivalence akumulátoru A a registru

Funkce PC = PC + 1
ACC = ACC xor Rn
Popis Mezi akumulátorem A a pracovním registrem Rn (maskou) je provedena logická funkce XOR. Výsledek je uložen do akumulátoru A, obsah pracovního registru se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 01101nnn (strojový kód = = 068H až 06FH dle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte XRL A,Rn

XRL A,direct	Nonekvivalence akumulátoru A a Direct

Funkce PC = PC + 2
ACC = ACC xor Direct
Popis Instrukce provede logickou funkci XOR mezi akumulátorem A a hodnotou Direct
Počet cyklů 1
Počet byte 2
Strojový kód 065H
Ovlivněné příznaky P

1. byte XRL A,direct
2. byte direct adr.

XRL direct,#data	Nonekvivalence hodnoty Direct a přímých dat

Funkce PC = PC + 3
Direct = Direct xor Data
Popis Instrukce provede logickou funkci XOR mezi hodnotou Direct a přímými daty
Počet cyklů 1
Počet byte 3
Strojový kód 063H
Ovlivněné příznaky žádné

1. byte XRL direct,#data
2. byte direct adr.
3. byte data

XRL direct,A	Nonekvivalence hodnoty Direct a akumulátoru A

Funkce PC = PC + 2
Direct = Direct xor ACC
Popis Instrukce provede funkci XOR mezi hodnotou Direct a akumulátorem A
Počet cyklů 1
Počet byte 2
Strojový kód 062H
Ovlivněné příznaky žádné

1. byte XRL direct,A
2. byte direct adr.

Bitové operace

Tyto operace se uskutečňují pouze s jednobitovými operandy. Pokud jako operand slouží některý z přímo adresovatelných bitů, je jeho adresa obsažena v dalším byte instrukce. Instrukce pracují buď s jedním operandem (SETB, CLR, CPL) nebo se dvěma operandy (ANL, ORL, XRL, MOV). Instrukce pracující se dvěma operandy používají jako jeden z operandů bit přenosu C. Kromě bitu C neovlivňují bitové instrukce žádné jiné příznaky.

ANL – logický součin bitů
ORL – logický součet bitů
CLR – nulování bitu
CPL – inverze bitu
MOV – přesun bitu
SETB – nastavení bitu

ANL C,bit	Logický součin C a bitu

Funkce PC = PC + 2
C = C and Bit
Popis Instrukce provede logický součin bitu přenosu C a bitu
Počet cyklů 2
Počet byte 2
Strojový kód 082H
Ovlivněné příznaky C
1. byte ANL C,bit
2. byte přímá adresa bitu

ANL C,/bit	Logický součin C s negovaným přímo adresovaným bitem

Funkce PC = PC + 2
C = C and Not Bit
Popis Instrukce provede logický součin AND bitu příznaku přenosu C s negovaným přímo adresovaným bitem bitové adresovatelné oblasti vnitřní datové paměti mikroprocesoru nebo bitově adresovatelného SFR. Výsledek je uložen do C. Přímá adresa bitu je součástí zápisu instrukce
Počet cyklů 2
Počet byte 2
Strojový kód 0B0H
Ovlivněné příznaky C

1. byte ANL C,/bit
2.byte přímá adresa bitu

CLR C	Nulování bitu přenosu C

Funkce PC = PC + 1
C = 0
Popis Bit přenosu může být po provedení některých instrukcí automaticky nastaven do 1. Touto instrukcí se dá bit přenosu nulovat
Počet cyklů 1
Počet byte 1
Strojový kód 0C3H
Ovlivněné příznaky C je nulován

1. byte CLR C

CLR bit	Nulování bitu

Funkce PC = PC + 2
Bit = 0
Popis Nulování bitu
Počet cyklů 1
Počet byte 2
Strojový kód 0C2H
Ovlivněné příznaky žádné

1. byte CLR bit
2. byte přímá adresa bitu

CPL C	Doplněk bitu přenosu C

Funkce PC = PC + 1
C = not C
Popis Současný stav bitu přenosu C je komplementován. Jednička se změní na nulu a naopak, nula se změní na jedničku
Počet cyklů 1
Počet byte 1
Strojový kód 0B3H
Ovlivněné příznaky C je komplementován

1. byte CPL C

CPL bit	Doplněk bitu

Funkce PC = PC + 2
Bit = not Bit
Popis Současný stav bitu je komplementován
Počet cyklů 1
Počet byte 2
Strojový kód 0B2H
Ovlivněné příznaky žádné

1. byte CPL bit
2. byte přímá adresa bitu

MOV C,bit	Přenos bitu do C

Funkce PC = PC + 2
C = Bit
Popis Instrukce přenese hodnotu bitu do C
Počet cyklů 1
Počet byte 2
Strojový kód 0A2H
Ovlivněné příznaky C

1. byte MOV C,bit
2. byte přímá adresa bitu

MOV bit,C	Přenos C do bitu

Funkce PC = PC + 2
Bit = C
Popis Instrukce přenese hodnotu C do bitu
Počet cyklů 2
Počet byte 2
Strojový kód 092H
Ovlivněné příznaky žádné

1. byte MOV bit,C
2. byte přímá adresa bitu

ORL C,bit	Logický součet bitu a příznaku přenosu C

Funkce PC = PC + 2
C = C or Bit
Popis Instrukce provede funkci OR s bitem a s příznakem přenosu C
Počet cyklů 2
Počet byte 2
Strojový kód 072H
Ovlivněné příznaky C

1. byte ORL C,bit
2. byte přímá adresa bitu

ORL C,/bit	Logický součet C a negovaného bitu

Funkce PC = PC + 2
C = C or not Bit
Popis Instrukce provede logický součet OR bitu příznaku přenosu C s negovaným přímoadresovaným bitem bitové adresovatelné oblasti vnitřní datové paměti mikropočítače nebo bitově adresovatelného SFR. Výsledek je uložen do C. Přímá adresa bitu je součástí zápisu instrukce
Počet cyklů 2
Počet byte 2
Strojový kód 0A0H
Ovlivněné příznaky C

1. byte ORL C,/bit
2. byte přímá adresa bitu

SETB C	Nastavení bitu příznaku C

Funkce PC = PC + 1
C = 1
Popis Instrukce nastaví bit příznaku C na hodnotu 1

Počet cyklů 1
Počet byte 1
Strojový kód 0D3H
Ovlivněné příznaky C je nastaven

1. byte SETB C

SETB bit	**Nastavení bitu**

Funkce PC = PC + 2
Bit = 1
Popis Instrukce nastaví bit na hodnotu 1
Počet cyklů 1
Počet byte 2
Strojový kód 0D2H
Ovlivněné příznaky žádné

1. byte SETB bit
2. byte přímá adresa bitu

Přenosy a výměny dat

Instrukce MOV a MOVX přenášejí hodnotu ze zdrojového operandu do cílového, přičemž se zdrojový operand nemění. Ostatní instrukce navzájem vyměňují hodnoty zdrojového a cílového operandu. Neovlivňují se žádné příznaky vyjma bitu P při práci s akumulátorem. Po vynulování (reset) mikropočítače je vybrána automaticky banka registrů 0.

MOV A,#data	**Přenos přímé hodnoty do akumulátoru A**

Funkce PC = PC + 2
ACC = Data
Popis Osmibitová hodnota určená přímým operandem Data se přenese do akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 074H
Ovlivněné příznaky P

1. byte MOV A,#data
2. byte data

Příklad: MOV A,#0FFH ; naplnění obsahu akumulátoru hodnotou 0FFH

MOV Rn,#data	**Přenos přímé hodnoty do registru**

Funkce PC = PC + 2
Rn = data
Popis Osmibitová hodnota určená přímým operandem Data se přenese do registru
Počet cyklů 1
Počet byte 2
Strojový kód 01111nnn (strojový kód = = 078H až 07FH dle použitého registru R0 až R7)
Ovlivněné příznaky žádné
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte MOV Rn,#data
2. byte data

Příklad: MOV R4,#0FFH ; naplnění obsahu registru R4 hodnotou 0FFH

MOV direct,#data	**Přenos přímé hodnoty do Direct**

Funkce PC = PC + 3
Direct = data
Popis Osmibitová hodnota určená přímým operandem Data se přenese do Direct
Počet cyklů 2
Počet byte 3
Strojový kód 075H
Ovlivněné příznaky žádné

1. byte MOV direct,#data
2. byte direct adr.
3. byte data

Příklad: MOV DPL,#50H ; naplnění DPL hodnotou 50H
MOV DPH,#00010110B ; naplnění DPH hodnotou 00010110B = 16H

MOV @Ri,#data	**Přenos přímé hodnoty do paměti dat**

Funkce PC = PC + 2
(Ri) = data
Popis Osmibitová hodnota určená přímým operandem Data se přenese do rezidentní paměti dat, která je naadresována obsahem registru Ri
Počet cyklů 1
Počet byte 2
Strojový kód 0111011i (strojový kód = = 076H až 077H dle použitého registru R0 až R1)
Ovlivněné příznaky žádné
Použitelné registry R0, R1

1. byte MOV @Ri,#data
2. byte (Ri) = data

Příklad: MOV R0,#3EH ; naplnění obsahu registru R0 adresou 3EH
MOV @R0,#85H ; 85H na adresu 3EH
INC R0 ; zvětšení obsahu registru R0 o jedničku
MOV @R0,#0CAH ; CAH na adresu 3FH

MOV DPTR,#data 16	**Přenos šestnáctibitových dat do registru DPTR**

Funkce PC = PC + 3
DPH = data (8 až 15)
DPL = data (0 až 7)
Popis Šestnáctibitová hodnota, určená přímým operandem Data 16, se přenese do registru DPTR
Počet cyklů 2
Počet byte 3
Strojový kód 090H
Ovlivněné příznaky žádné

1. byte MOV DPTR,#data 16
2. byte data (8 až 15)
3. byte data (0 až 7)

Příklad: MOV DPTR,#0FFFFH ; naplnění DPTR šestnáctibitovou hodnotou FFFFH

MOV A,Rn	**Přenos obsahu registru do akumulátoru A**

Funkce PC = PC + 1
ACC = Rn
Popis Obsah pracovního registru (R0 až R7) zvolené banky se přenese do akumulátoru A
Počet cyklů 1
Počet byte 1
Strojový kód 11101nnn (strojový kód = = 0E8H až 0EFH dle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte MOV A,Rn

Příklad: MOV R3,#30H ; hodnota 30H do R0
MOV A,R3 ; přenos hodnoty 30H do A, hodnota 30H zůstává i v R3

MOV A,direct	**Přenos hodnoty Direct do akumulátoru A**

Funkce PC = PC + 2
ACC = Direct
Popis Osmibitová hodnota určená Direct se přenese do akumulátoru A
Počet cyklů 1
Počet byte 2
Strojový kód 0E5H
Ovlivněné příznaky P

1. byte MOV A,direct
2. byte direct adr.

Příklad: MOV A,DPL ; přenos hodnoty z DPL do akumulátoru

MOV A,@Ri	**Přenos z rezidentní paměti dat do akumulátoru A**

Funkce PC = PC + 1
ACC = (Ri)
Popis Obsah místa rezidentní paměti dat naadresované registrem Ri se přenese do akumulátoru A. Obsah registru Ri a paměťového místa dat se nezmění
Počet cyklů 1
Počet byte 1
Strojový kód 1110011i (strojový kód = = 0E6H až 0E7H dle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1

1. byte MOV A,@Ri

Příklad: MOV R1,#37H ; naplnění obsahu R1 adresou 37H
MOV A,@R1 ; přenos z adresy 37H do akumulátoru

MOV Rn,A	**Přenos hodnoty akumulátoru A do pracovního registru**

Funkce PC = PC + 1
Rn = ACC
Popis Obsah akumulátoru A se přenáší do registru Rn zvolené banky registrů
Počet cyklů 1
Počet byte 1
Strojový kód 11111nnn (strojový kód = = 0F8H až 0FFH dle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte MOV Rn,A

Příklad: MOV A,#37H ; naplnění akumulátoru hodnotou 37H
MOV R6,A ; přenos hodnoty 37H z akumulátoru do R6

MOV Rn,direct	**Hodnota Direct se přenese do akumulátoru A**

Funkce PC = PC + 2
Rn = Direct
Popis Obsah osmibitové hodnoty Direct se přenese do akumulátoru A
Počet cyklů 2
Počet byte 2
Strojový kód 10101nnn (strojový kód = = 0A8H až 0AFH dle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte MOV Rn,direct
2. byte direct adr.

Příklad: MOV R4,DPH ; přenos z DPH do registru R4

MOV direct,A	**Hodnota akumulátoru A se přenese do Direct**

Funkce PC = PC + 2
Direct = Acc
Popis Osmibitová hodnota v akumulátoru se přenese do Direct
Počet cyklů 1
Počet byte 2
Strojový kód 0F5H
Ovlivněné příznaky P

1. byte MOV direct,A
2. byte direct adr.

Příklad: MOV DPH,A ; přenos z akumulátoru do DPH

MOV direct,Rn	**Obsah registru Rn se přenese do Direct**

Funkce PC = PC + 2
Direct = Rn
Popis Obsah pracovního registru Rn zvolené banky se přenese do Direct
Počet cyklů 2

Počet byte 2
Strojový kód 10001nnn (strojový kód = = 088H až 08FH dle použitého registru R0 až R7)
Ovlivněné příznaky P
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7
1. byte MOV direct,Rn
2. byte direct adr.

MOV irect,@Ri — Obsah rezidentní paměti dat se přenese do Direct

Funkce PC = PC + 2
Direct = (Ri)
Popis Obsah místa rezidentní paměti dat adresované registrem Ri se přenese do Direct
Počet cyklů 2
Počet bytů 2
Strojový kód 1000011i (strojový kód = = 086H až 087H dle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1
1. byte MOV direct,@Ri
2. byte direct adr.

MOV @Ri,A — Přenos obsahu akumulátoru A do rezidentní paměti dat

Funkce PC = PC + 1
(Ri) = ACC
Popis Obsah místa rezidentní paměti dat se přenese do akumulátoru A. Obsah registru Ri a paměťového místa se nezmění
Počet cyklů 1
Počet byte 1
Strojový kód 1111011i (strojový kód = = 0F6H až 0F7H dle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1
1. byte MOV @Ri,a

MOV @Ri,direct — Přenos Direct do rezidentní paměti dat

Funkce PC = PC + 1
(Ri) = direct
Popis Obsah Direct se přenese do rezidentní paměti dat
Počet cyklů 1
Počet byte 1
Strojový kód 1010011i (strojový kód = = 0A6H až 0A7H dle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1
1. byte MOV @Ri,direct

MOV direct,direct — Přenos Direct do Direct

Funkce Direct 1 = Direct 2
Popis Přenos osmibitové hodnoty jednoho paměťového místa Direct 1 do druhého Direct 2
Počet cyklů 2
Počet byte 3
Strojový kód 085H
Ovlivněné příznaky žádné
1. byte MOV direct1 = direct 2
2. byte direct adr. 2
3. byte direct adr. 1
Příklad: MOV DPH,DPL ; přenos z DPH do DPL

MOVC A,@A+DPTR — Přenos bytu z určené adresy do akumulátoru

Funkce PC = PC + 1
ACC = code (ACC + DPTR)
Popis Instrukce naadresuje paměť programu adresou vzniklou sečtením DPTR a ACC a obsah bytu na této adrese přenese do akumulátoru. Tímto způsobem lze z celé paměti programu číst zadané konstanty
Počet cyklů 2
Počet byte 1
Strojový kód 083H
Ovlivněné příznaky P
1. byte MOVC A,@A+DPTR
Příklad:
```
MOV A,#0          ; hodnota 0 do akumulátoru
MOV DPL,#0FEH     ; hodnota 0FEH do DPL
MOV DPH,#0        ; hodnota 0 do DPH, nyní je DPTR = 00FEH
MOVC A,@A+DPTR    ; obsah bytu z adresy FEH do akumulátoru
INC A             ; přičtení 1 k hodnotě akumulátoru
MOVC A,@A+DPTR    ; obsah bytu z adresy FFH do akumulátoru
```

MOVX A,@Ri — Přenos z vnější paměti dat do akumulátoru A

Funkce PC = PC + 1
ACC = ext. paměť (Ri)
Popis Instrukce přenese obsah místa v externí paměti dat adresovaného re gistrem R0 nebo R1 do akumulátoru A. Při tomto způsobu adresování je k dispozici prostor 256 B externí paměti, větší rozsah lze zajistit např. stránkováním
Počet cyklů 2
Počet byte 1
Strojový kód 1110001i (strojový kód = = 0E2H až 0E3H dle použitého registru R0 až R1)
Ovlivněné příznaky žádné
Použitelné registry R0, R1
1. byte MOVX A,@Ri
Příklad:
```
MOV R1,#37H   ; naplnění obsahu R1 adresou 37H
MOVX A,@Ri    ; přenos obsahu paměti dat s adresou 37H do akumulátoru
```

MOVX @Ri,A — Přenos z akumulátoru A do vnější paměti dat

Funkce PC = PC + 1
ext. paměť (Ri) = ACC
Popis Instrukce přenese obsah akumulátoru A do externí paměti dat adresovaného registrem R0 nebo R1. Při tomto způsobu adresování je k dispozici prostor 256 B externí paměti, větší rozsah lze zajistit např. stránkováním
Počet cyklů 2
Počet byte 1
Strojový kód 1111001i (strojový kód = = 0F2H až 0F3H dle použitého registru R0 až R1)
Ovlivněné příznaky žádné
Použitelné registry R0, R1
1. byte MOVX @Ri,A
Příklad:
```
MOV R1,#37H    ; naplnění obsahu R1 adresou 37H
MOV A,#0FFH    ; naplnění akumulátoru hodnotou FFH
MOVX @Ri,A     ; přenos hodnoty FFH na adresu 37H paměti dat
```

MOVX @DPTR,A — Přenos z akumulátoru A do vnější paměti dat

Funkce PC = PC + 1
ext. paměť (DPTR) = ACC
Popis Instrukce přenese obsah akumulátoru A do externí paměti dat adresovaného registrem DPTR. Při tomto způsobu adresování je k dispozici celý prostor 64 kB externí paměti
Počet cyklů 2
Počet byte 1
Strojový kód 0F0H
Ovlivněné příznaky žádné
1. byte MOVX @DPTR,A

MOVX A,@DPTR — Přenos z vnější paměti dat do akumulátoru

Funkce PC = PC + 1
ACC = ext. paměť (DPTR)
Popis Instrukce přenese obsah místa v externí paměti dat adresovaného registrem DPTR do akumulátoru A. Při tomto způsobu adresování je k dispozici celý prostor 64 kB externí paměti
Počet cyklů 2
Počet byte 1
Strojový kód 0E0H
Ovlivněné příznaky žádné
1. byte MOVX A,@DPTR

XCH A,Rn — Výměna obsahů akumulátoru A a pracovního registru

Funkce PC = PC + 1
ACC ←→ Rn
Popis Vymění se obsahy akumulátoru A a pracovního registru Rn
Počet cyklů 1
Počet byte 1
Strojový kód 11001nnn (strojový kód = = 0C8H až 0CFH dle použitého registru R0 až R7)
Ovlivněné příznaky P
1. byte XCH A,Rn
Příklad: přesun PSW do registru R5 bez ztráty obsahu akumulátoru
```
XCH A,R5    ; výměna obsahu akumulátoru a registru R5
XCH A,PSW   ; výměna obsahu akumulátoru a PSW
XCH A,R5    ; výměna obsahu akumulátoru a registru R5
```

XCH A,direct — Výměna obsahů akumulátoru A a Direct

Funkce PC = PC + 2
ACC ←→ direct
Popis Vymění se obsahy akumulátoru A a hodnoty v Direct
Počet cyklů 1
Počet byte 2
Strojový kód 0C5H
Ovlivněné příznaky P
1. byte XCH A,direct
2. byte direct adr.
Příklad:
```
XCH A,DPL   ; výměna obsahů akumulátoru a DPL
```

XCH A,@Ri — Výměna obsahu akumulátoru A a adresy v paměti dat

Funkce PC = PC + 1
ACC ←→ (Ri)
Popis Obsah akumulátoru A a obsah adresového místa v paměti dat se vymění. Obsah registru Ri se nemění
Počet cyklů 1
Počet byte 1
Strojový kód 1100011i (strojový kód = = 0C6H až 0C7H dle použitého registru R0 až R1)
Ovlivněné příznaky P
Použitelné registry R0, R1
1. byte XCH A,@Ri
Příklad: snížení obsahu dat na zvolené adrese o jedničku
```
MOV R0,#26H   ; adresa 26H do R0
XCH A,@R0     ; výměna obsahu dat na adrese 26 s akumulátorem
DEC A         ; snížení obsahu akumulátoru o 1
XCH A,@R0     ; výměna obsahu dat na adrese 26H s akumulátorem
```

XCHD A,@Ri — Výměna 4 bitů mezi akumulátorem a pamětí dat

Funkce PC = PC + 1
ACC (bity 0 až 3) ←→ (Ri) (bity 0 až 3)

Popis	Instrukce vymění bity 0 až 3 akumulátoru A s bity 0 až 3 obsahu adresy paměti dat nepřímo adresovaného pomocí registru R0 nebo R1
Počet cyklů	1
Počet byte	1
Strojový kód	1101011i (strojový kód = = 0D6H až 0D7H dle použitého registru R0 až R1)
Ovlivněné příznaky	P
Použitelné registry	R0, R1
1. byte	XCHD A,@Ri

Příklad:
```
MOV R0,#26H   ; adresa 26H do R0
CLR A         ; nulování akumulátoru
XCHD A,@R0    ; výměna čtyř nižších bitů akumulátoru se čtyřmi nižšími bity obsahu dat na adrese 26H
```

SWAP A	Výměna čtyř bitů uvnitř akumulátoru A
Funkce	PC = PC + 1 ACC (0 až 3) ← → ACC (4 až 7)
Popis	Instrukce vymění navzájem nižší a vyšší čtyři bity v akumulátoru A
Počet cyklů	1
Počet byte	1
Strojový kód	0C4H
Ovlivněné příznaky	P
1. byte	SWAP A

Příklad: Sestavení osmibitového slova, složeného z bitů 0 až 3 dvou slov, uložených na adresách 32H a 33H. Výsledné slovo se uloží na adresu 33H
```
MOV R0,#32H   ; adresa 32H do pracovního registru R0
MOV R1,#33H   ; adresa 33H do pracovního registru R1
XCHD A,@R0    ; výměna čtyř nižších bitů akumulátoru a obsahu adr. 32H
SWAP A        ; výměna čtveřic bitů v akumulátoru
XCHD A,@R1    ; výměna čtyř nižších bitů akumulátoru a obsahu adr. 33H
MOV @R0,A     ; přesun obsahu akumulátoru na adresu 32H
```

Čítání nahoru a dolů

Instrukce přičítají nebo odečítají jedničku. Při přetečení nebo podtečení operandu se negeneruje žádný příznak.

INC - increment - zvýšení hodnoty operandu o 1

DEC - decrement - snížení hodnoty operandu o 1

DEC A	Snížení obsahu akumulátoru A o 1
Funkce	PC = PC + 1 ACC = ACC - 1
Popis	Obsah akumulátoru A se sníží o jedničku. Instrukce nenastavuje kromě parity žádné příznaky (např. C)
Počet cyklů	1
Počet byte	1
Strojový kód	014H
Ovlivněné příznaky	P
1. byte	DEC A

DEC Rn	Snížení obsahu pracovního registru o 1
Funkce	PC = PC + 1 Rn = Rn - 1
Popis	Obsah pracovního registru se sníží o jedničku
Počet cyklů	1
Počet byte	1
Strojový kód	00011nnn (strojový kód = = 018H až 01FH dle použitého registru R0 až R7)
Ovlivněné příznaky	žádné
Použitelné registry	R0, R1, R2, R3, R4, R5, R6, R7
1. byte	DEC Rn

DEC @Ri	Snížení obsahu adresy rezidentní paměti dat o 1
Funkce	PC = PC + 1 (Ri) = (Ri) -1
Popis	Obsah adresy rezidentní paměti dat se sníží o jedničku
Počet cyklů	1
Počet byte	1
Strojový kód	0001011i (strojový kód = = 016H až 017H dle použitého registru R0 až R1)
Ovlivněné příznaky	žádné
Použitelné registry	R0, R1
1. byte	DEC @Ri

DEC direct	Snížení obsahu Direct o 1
Funkce	PC = PC + 2 direct = direct - 1
Popis	Obsah Direct se sníží o jedničku
Počet cyklů	1
Počet byte	2
Strojový kód	015H
Ovlivněné příznaky	žádné
1. byte	DEC direct
2. byte	direct adr.

INC A	Zvýšení obsahu akumulátoru A o 1
Funkce	PC = PC + 1 ACC = ACC + 1
Popis	Obsah akumulátoru A se zvýší o 1
Počet cyklů	1
Počet byte	1
Strojový kód	04H
Ovlivněné příznaky	P
1. byte	INC A

INC Rn	Zvýšení obsahu pracovního registru o 1
Funkce	PC = PC + 1 Rn = Rn + 1
Popis	Obsah pracovního registru se zvýší o jedničku
Počet cyklů	1
Počet byte	1
Strojový kód	00001nnn (strojový kód = = 08H až 0FH dle použitého registru R0 až R7)
Ovlivněné příznaky	žádné
Použitelné registry	R0, R1, R2, R3, R4, R5, R6, R7
1. byte	INC Rn

INC @Ri	Zvýšení obsahu adresy rezidentní paměti dat o 1
Funkce	PC = PC + 1 (Ri) = (Ri) + 1
Popis	Obsah adresy rezidentní paměti dat se zvýší o jedničku
Počet cyklů	1
Počet byte	1
Strojový kód	0000011i (strojový kód = = 06H až 07H dle použitého registru R0 až R1)
Ovlivněné příznaky	žádné
Použitelné registry	R0, R1
1. byte	INC @Ri

INC direct	Zvýšení obsahu Direct o 1
Funkce	PC = PC + 2 direct = direct + 1
Popis	Obsah Direct se zvýší o jedničku
Počet cyklů	1
Počet byte	2
Strojový kód	05H
Ovlivněné příznaky	žádné
1. byte	INC direct
2. byte	direct adr.

INC DPTR	Zvýšení obsahu registrového páru DPTR o 1
Funkce	PC = PC + 1 DPTR = DPTR + 1
Popis	Registrový šesnáctibitový pár DPTR se skládá ze dvou osmibitových registrů DPL a DPH. DPL určuje spodní část a DPH horní část DPTR. Instrukcí se zvýší hodnota celého DPTR o 1
Počet cyklů	2
Počet byte	1
Strojový kód	0A3H
Ovlivněné příznaky	žádné
1. byte	INC DPTR

Rotace

Rotace, čili kruhový posun bitů akumulátoru A, je zastoupena čtyřmi instrukcemi. Rotace může být vpravo nebo vlevo a s účastí bitu přenosu C či nikoliv. V podstatě jde o to, že při rotaci vpravo bez účasti bitu přenosu C se všechny bity osmibitového slova v akumulátoru posunou o jednu pozici vpravo, přičemž bit 0 přejde na pozici bitu 7. Obdobně při rotaci vlevo bez účasti bitu přenosu C se všechny bity osmibitového slova v akumulátoru posunou o jednu pozici vlevo, přičemž bit 7 přejde na pozici bitu 0.

Při rotaci vpravo s účastí bitu přenosu C se všechny bity osmibitového slova v akumulátoru posunou o jednu pozici vpravo, bit 0 přejde do C a hodnota v C přejde do bitu 7 akumulátoru. Při rotaci vlevo s účastí bitu přenosu C se všechny bity osmibitového slova v akumulátoru posunou o jednu pozici vlevo, bit 7 přejde do C a hodnota v C přejde do bitu 0 akumulátoru.

RL - rotace v akumulátoru o jednu pozici vlevo

RLC - rotace v akumulátoru o jednu pozici vlevo přes bit C

RR - rotace v akumulátoru o jednu pozici vpravo

RRC - rotace v akumulátoru o jednu pozici vpravo přes bit C

RL A	Rotace akumulátoru vlevo
Funkce	PC = PC + 1 ACC = RL A
Popis	Obsah akumulátoru se posune o jeden bit vlevo, bit 7 přejde do bitu 0
Počet cyklů	1
Počet byte	1
Strojový kód	023H
Ovlivněné příznaky	žádné
1. byte	RL A

Příklad:
```
MOV A,#10110010B   ; akumulátor obsahuje 01011001B
RL A               ; akumulátor obsahuje 10110010B
```

RLC A	Rotace akumulátoru vlevo přes C
Funkce	PC = PC + 1 ACC = RLC A
Popis	Obsah akumulátoru se posune o jeden bit vlevo, bit 7 přejde do C a hodnota v C do bitu 0
Počet cyklů	1
Počet byte	1
Strojový kód	033H
Ovlivněné příznaky	žádné
1. byte	RLC A

RR A	Rotace akumulátoru vpravo
Funkce	PC = PC + 1 ACC = RR A
Popis	Obsah akumulátoru se posune o jeden bit vpravo, bit 0 přejde do bitu 7
Počet cyklů	1
Počet byte	1
Strojový kód	03H
Ovlivněné příznaky	žádné
1. byte	RR A

RRC A	Rotace akumulátoru vpravo přes C
Funkce	PC = PC + 1 ACC = RRC A
Popis	Obsah akumulátoru se posune o jeden bit vpravo, bit 0 přejde do C a hodnota v C do bitu 7
Počet cyklů	1
Počet byte	1
Strojový kód	013H
Ovlivněné příznaky	žádné
1. byte	RRC A

Operace se zásobníkem, podprogramy

Uvedené instrukce pracují se zásobníkem, s oblastí vnitřní datové paměti, která je nepřímo adresovaná pomocí registru SP.

ACALL – volání podprogramu v rámci stránky

LCALL – volání podprogramu dlouhé

PUSH – uložení hodnoty do zásobníku

POP – vytažení hodnoty ze zásobníku

RET – návrat z podprogramu

RETI – návrat z podprogramu přerušení

CALL	Volání podprogramu
Popis	Tato instrukce není součástí instrukcí „51“, avšak assembler sám podle adresy přiřadí instrukci LCALL nebo ACALL

ACALL	Volání podprogramu v rámci stránky
Funkce	PC = PC + 2 SP = SP + 1 (SP) = PC (0 až 7) SP = SP + 1 (SP) = PC (8 až 15) PC = A (0 až 10) + PC (11 až 15)
Popis	Návratová adresa se uloží do zásobníku, cílová adresa se vytvoří tak, že bity PC (11 až 15) zůstanou zachovány a bity PC (0 až 10) se nahradí adresou A (0 až 10), kterou obsahuje zápis instrukce. Touto instrukcí lze tedy volat podprogram pouze uvnitř jedné stránky 2 kB
Počet cyklů	2
Počet byte	2
Strojový kód	011H, 031H, 051H, 071H, 091H, 0B1H, 0D1H, 0F1H
Ovlivněné příznaky	žádné
1. byte	ACALL
2. byte	operand (nižší byte adresy)

LCALL	Volání podprogramu dlouhé
Funkce	PC = PC + 2 SP = SP + 1 (SP) = PC (0 až 7) SP = SP + 1 (SP) = PC (8 až 15) PC = šestnáctibitová adresa podprogramu
Popis	Návratová adresa se uloží do zásobníku. Programový čítač je potom naplněn novou adresou, která je obsažena v zápisu instrukce. Program potom dále pokračuje na nové adrese
Počet cyklů	2
Počet byte	3
Strojový kód	012H
Ovlivněné příznaky	žádné
1. byte	LCALL
2. byte	vyšší byte adresy podprogramu
3. byte	nižší byte adresy podprogramu

PUSH direct	Uložení do zásobníku
Funkce	PC = PC + 2 SP = SP + 1 (SP) = direct
Popis	Instrukce uloží do zásobníku obsah přímo adresovaného bytu vnitřní paměti dat nebo SFR. Přímá adresa je součástí zápisu instrukce
Počet cyklů	2
Počet byte	2
Strojový kód	0C0H
Ovlivněné příznaky	žádné
1. byte	PUSH direct
2. byte	direct adr.
Příklad: PUSH ACC	; uloží obsah akumulátoru do zásobníku

POP direct	Výběr ze zásobníku
Funkce	PC = PC + 2 direct = (SP) SP = SP - 1
Popis	Instrukce vybere ze zásobníku byte a uloží jej do přímo adresovaného byte vnitřní paměti dat nebo SFR. Přímá adresa je součástí zápisu instrukce
Počet cyklů	2
Počet byte	2
Strojový kód	0D0H
Ovlivněné příznaky	žádné
1. byte	POP direct
2. byte	direct adr.
Příklad: POP ACC	; vybere obsah zásobníku a uloží do akumulátoru

RET	Návrat z podprogramu
Funkce	PC (8 až 15) = (SP) SP = SP – 1 PC (0 až 7) = (SP) SP = SP - 1
Popis	Instrukce vybere ze zásobnímu šestnáctibitovou návratovou adresu a uloží jí do programového čítače. Program potom pokračuje na této adrese
Počet cyklů	2
Počet byte	1
Strojový kód	022H
Ovlivněné příznaky	žádné
1. byte	RET

RETI	Návrat z podprogramu s obnovou příznaků
Funkce	PC (8 až 15) = (SP) SP = SP – 1 PC (0 až 7) = (SP) SP = SP - 1
Popis	Instrukce vybere ze zásobnímu šestnáctibitovou návratovou adresu a uloží jí do programového čítače. Program potom pokračuje na této adrese. Zároveň se uvolní přerušovací logika a nadále může být akceptováno další přerušení
Počet cyklů	2
Počet byte	1
Strojový kód	032H
Ovlivněné příznaky	žádné
1. byte	RETI

Nepodmíněné skoky

Nepodmíněné skoky umožňují pohyb v progrmu nezávisle na jakýchkoliv pomínkách.

AJMP - skok na absolutní adresu a v rámci stránky

JMP - relativní skok podle ACC + DPTR

LJMP - dlouhý skok na absolutní adresu

SJMP - krátký skok na relativní adresu

AJMP	Skok na absolutní adresu v rámci stránky
Funkce	PC = PC + 2 PC = A (0 až 10) + PC (11 až 15)
Popis	Instrukce provede skok na adresu, která se vytvoří tak, že bity PC (11 až 15) zůstanou zachovány a bity PC (0 až 10) se nahradí adresou A (0 až 10), kterou obsahu je zápis instrukce. Touto instrukcí lze tedy skákat pouze uvnitř jedné stránky 2 kB
Počet cyklů	2
Počet byte	2
Strojový kód	01H, 21H, 41H, 61H, 81H, A1H, C1H, E1H
Ovlivněné příznaky	žádné
1. byte	AJMP
2. byte	operand (nižší byte adresy)

JMP @A + DPTR	Skok dle ACC + DPTR
Funkce	PC = ACC + DPTR
Popis	Programový čítač je naplněn šestnáctibitovou adresou, vzniklou sečtením obsahu akumulátoru A a registrového páru DPTR. Program dále pokračuje na nové adrese.
Počet cyklů	2
Počet byte	1
Strojový kód	073H
Ovlivněné příznaky	žádné
1. byte	JMP @A + DPTR

JMP	Skok
Popis	Tato instrukce není součástí instrukcí „51“, avšak assembler sám podle adresy přiřadí instrukci LJMP, SJMP nebo AJMP.

LJMP	Dlouhý skok
Funkce	PC (0 až 7) = Adr. (0 až 7) PC (8 až 15) = Adr (0 až 15)
Popis	Programový čítač je naplněn adresou, která je celá obsažena v zápisu instrukce. Program dále pokračuje na nové adrese
Počet cyklů	2
Počet byte	3
Strojový kód	02H
Ovlivněné příznaky	žádné
1. byte	LJMP
2. byte	vyšší byte adresy
3. byte	nižší byte adresy

SJMP	Krátký skok
Funkce	PC = PC + 2 PC = PC + relativní posun
Popis	Programový čítač se nejprve nastaví na adresu následující instrukce. Potom se k PC přičte relativní posun, který je součástí zápisu instrukce a který může nabývat hodnot -128 až +127. Program dále pokračuje na nové adrese
Počet cyklů	2
Počet byte	2
Strojový kód	080H
Ovlivněné příznaky	žádné
1. byte	SJMP
2. byte	relativní posun

Podmíněné skoky

Podmíněné skoky jsou takové, které potřebují k vykonání posunu PC nějakou podmínku. Např. instrukce CJNE (compare and jump if not equal) = porovnej, a pokud hodnota není shodná, skoč. Všechny instrukce podmíněných skoků používají pouze relativní adresu. Výsledná adresa vychází z momentálního stavu PC a z offsetu, který je součástí zápisu instrukce a může nabývat hodnot -128 až +127. Jedinou instrukcí, která ovlivňuje příznaky, je CJNE.

CJNE A,direct	V případě nerovnosti operandů akumulátoru A a Direct skoč
Funkce	PC = PC + 3 if ACC < direct then PC = PC + relativní posun; C = 1 if ACC > direct then PC = PC + relativní posun; C = 0 if ACC = direct potom program pokračuje na následující adrese
Popis	Instrukce provede komparaci operandů ACC a Direct a provede případný skok. V případě rovnosti ope-

randů program pokračuje na následující adrese. Pokud je ACC menší než Direct, je nastaven příznak C a program skočí na danou adresu. Pokud je ACC větší než Direct, je nulován příznak C a program skočí na danou adresu

Počet cyklů 2
Počet byte 3
Strojový kód 0B5H
Ovlivněné příznaky C

1. byte CJNE A,direct
2. byte direct adr.
3. byte relativní posun (-128 až +127)

CJNE A,#data	**V případě nerovnosti operandů akumulátoru A a přímých dat skoč**

Funkce PC = PC + 3
if ACC < data then PC = PC + relativní posun; C = 1
if ACC > data then PC = PC + relativní posun; C = 0
if ACC = data potom program pokračuje na následující adrese

Popis Instrukce provede komparaci hodnot v akumulátoru A a přímých dat a provede případný skok. V případě rovnosti hodnot program pokračuje na následující adrese. Pokud je ACC menší než data, je nastaven příznak C a program skočí na danou adresu. Pokud je ACC větší než data, je nulován příznak C a program skočí na danou adresu

Počet cyklů 2
Počet byte 3
Strojový kód 0B4H
Ovlivněné příznaky C

1. byte CJNE A,#data
2. byte data
3. byte relativní posun (-128 až +127)

CJNE @Ri,#data	**V případě nerovnosti nepřímých a přímých dat skoč**

Funkce PC = PC + 3
if (Ri) < data then PC = PC + relativní posun; C = 1
if (Ri) > data then PC = PC + relativní posun; C = 0
if (Ri) = data potom program pokračuje na následující adrese

Popis Instrukce provede komparaci nepřímých a přímých dat a provede případný skok. V případě rovnosti operandů program pokračuje na následující adrese. Pokud jsou nepřímá data menší než data přímá, je nastaven příznak C a program skočí na danou adresu. Pokud jsou nepřímá data větší než data přímá, je nulován příznak C a program skočí na danou adresu. Přímá data jsou součástí zápisu instrukce

Počet cyklů 2
Počet byte 3
Strojový kód 1011011i (strojový kód = = 0B6H až 0B7H dle použitého registru R0 až R1)
Ovlivněné příznaky C
Použitelné registry R0, R1

1. byte CJNE @Ri,#data
2. byte data
3. byte relativní posun (-128 až +127)

CJNE Rn,#data	**V případě nerovnosti operandů registru a přímých dat skoč**

Funkce PC = PC + 3
if Rn < data then PC = PC + relativní posun; C = 1
if Rn > data then PC = PC + relativní posun; C = 0
if Rn = data potom program pokračuje na následující adrese

Popis Instrukce provede komparaci pracovního registru a přímých dat a provede případný skok. V případě rovnosti operandů program pokračuje na následující adrese. Pokud je hodnota pracovního registru menší než hodnota přímých dat, je nastaven příznak C a program skočí na danou adresu. Pokud je hodnota pracovního registru větší než hodnota přímých dat, je nulován příznak C a program skočí na danou adresu. Přímá data jsou součástí zápisu instrukce

Počet cyklů 2
Počet byte 3
Strojový kód 10111nnn (strojový kód = = 0B8H až 0BFH dle použitého registru R0 až R7)
Ovlivněné příznaky C
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte CJNE Rn,#data
2. byte data
3. byte relativní posun (-128 až +127)

DJNZ Rn	**Snížení obsahu registru o 1 a skok, není-li hodnota v registru nulová**

Funkce PC = PC + 2
Rn = Rn – 1
if Rn <> 0 then PC = PC + rel. posun

Popis Nejprve se sníží opearnd registru o 1, otestuje se, a v případě nenulového výsledku skočí na danou adresu

Počet cyklů 2
Počet byte 2
Strojový kód 11011nnn (strojový kód = = 0D8H až 0DFH dle použitého registru R0 až R7)
Ovlivněné příznaky žádné
Použitelné registry R0, R1, R2, R3, R4, R5, R6, R7

1. byte DJNZ Rn
2. byte relativní posun

DJNZ direct	**Snížení obsahu Direct o 1 a skok, není-li hodnota v Direct nulová**

Funkce PC = PC + 3
Direct = Direct – 1
if Direct <> 0 then PC = PC + relativní posun

Popis Nejprve se sníží opearnd Direct o 1, otestuje se a v případě nenulového výsledku skočí na danou adresu

Počet cyklů 2
Počet byte 3
Strojový kód 0D5H
Ovlivněné příznaky žádné

1. byte DJNZ direct
2. byte direct adr.
3. byte relativní posun

JB	**Pokud je bit = 1, skok**

Funkce PC = PC + 3
if Bit = 1 then PC = PC + rel. posun

Popis Instrukce testuje bit z bitově adresovatelné oblasti vnitřní RAM nebo SFR. Skok, pokud bit = 1, jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 3
Strojový kód 020H
Ovlivněné příznaky žádné

1. byte JB
2. byte adresa bitu
3. byte relativní posun

JBC	**Pokud je bit = 1, skok a nulování bitu**

Funkce PC = PC + 3
if Bit = 1 then PC = PC + relativní posun; Bit = 0

Popis Instrukce testuje bit z bitové oblasti vnitřní RAM nebo SFR. Skok, pokud bit = 1, a zároveň nulování bitu. Jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 3
Strojový kód 010H
Ovlivněné příznaky žádné

1. byte JBC
2. byte adresa bitu
3. byte relativní posun

JC	**Pokud je C = 1, skok**

Funkce PC = PC + 2
if C = 1 then PC = PC + rel. posun

Popis Instrukce testuje bit přenosu C. Skok, pokud bit = 1. Jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 2
Strojový kód 040H
Ovlivněné příznaky žádné

1. byte JC
2. byte relativní posun

JNB	**Pokud je bit = 0, skok**

Funkce PC = PC + 3
if Bit = 0 then PC = PC + rel. posun

Popis Instrukce testuje bit z bitové oblasti vnitřní RAM nebo SFR. Skok, pokud bit = 0. Jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 3
Strojový kód 030H
Ovlivněné příznaky žádné

1. byte JNB
2. byte adresa bitu
3. byte relativní posun

JNC	**Pokud je C = 0, skok**

Funkce PC = PC + 2
if C = 0 then PC = PC + rel. posun

Popis Instrukce testuje bit přenosu C. Skok, pokud bit = 0. Jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 2
Strojový kód 050H
Ovlivněné příznaky žádné

1. byte JNC
2. byte relativní posun

JZ	**Pokud je hodnota akumulátotu A = 0, skok**

Funkce PC = PC + 2
if ACC = 0 then PC = PC + rel. posun

Popis Instrukce testuje akumalátor A. Skok, pokud ACC = 0. Jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 2
Strojový kód 060H
Ovlivněné příznaky žádné

1. byte JZ
2. byte relativní posun

JNZ	**Pokud je hodnota akumulátotu A různá od nuly, skok**

Funkce PC = PC + 2
if ACC <> 0 then PC = PC + rel. posun

Popis Instrukce testuje akumulátor A. Skok, pokud hodnota ACC není nulová. Jinak program pokračuje na následující adrese

Počet cyklů 2
Počet byte 2
Strojový kód 070H
Ovlivněné příznaky žádné

1. byte JNZ
2. byte relativní posun